S00P0089US00
09/647265

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT :
APPLICATION No. : Group Art Unit :
FILING DATE : Examiner :
TITLE :

Hon. Commissioner of Patents and Trademarks,
Washington, D.C. 20231

SIR:

CERTIFIED TRANSLATION

I, Mihoko Takeda, am an official translator of the Japanese language into the English language and I hereby certify that the attached comprises an accurate translation into English of Japanese Application No. 11-023068, filed on January 29, 1999.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

September 14, 2000
Date

Mihoko Takeda
Mihoko Takeda

Patent Office

Japanese Government

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: January 29, 1999

Application Number: Patent Application

Ser. No. 11-023068

Applicant: Sony Corporation

November 19, 1999

Commissioner,

Patent Office Takahiko Kondo

[Document Name]	Patent Application
[Reference Number]	9900035603
[To]	Hon.Commissioner, Patent Office
[IPC]	G01B 11/24
[Inventor]	
[Address]	c/o Sony Corporation 7-35, Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan
[Name]	Toby Walker
[Inventor]	
[Address]	c/o Sony Corporation 7-35, Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan
[Name]	Hiroshi Matsubara
[Patent Applicant]	
[Identification Number]	000002185
[Name]	Sony Corporation
[Representative]	Nobuyuki Idei
[Patent Attorney]	
[Identification Number]	100067736
[Patent Attorney]	
[Name]	Akira Koike
[Patent Attorney]	
[Identification Number]	100086335
[Patent Attorney]	
[Name]	Eiichi Tamura
[Patent Attorney]	
[Identification Number]	100096677
[Patent Attorney]	

[Name]	Seiji Iga	
[Indication of Charge]		
[Number of Prepaid Ledger]	019530	
[Amount]	21,000 yen	
[List of Document]		
[Document]	Specification	1
[Document]	Drawing	1
[Document]	Summary	1
[General Power of Attorney Number]		9707387
[Need of Proof]	Yes	

[Document] Specification

[Title of the Invention]

DATA DESCRIBING METHOD, AND DATA PROCESSOR

[What is claimed is:]

[Claim 1]

A data describing method for describing feature data representing characteristics of contents of input data, wherein

a descriptor becoming a unit element defining a describing method includes structure to have its attribute defined as a lower-level element and may include a descriptor as the attribute.

[Claim 2]

The data describing method according to claim 1, wherein

the input data includes at least one of visual data or audio data included in video data.

[Claim 3]

The data describing method according to claim 1, wherein

the descriptor is generated to inherit functions including a capability of forming the structure from a higher-class descriptor .

[Claim 4]

The data describing method according to Claim 1, wherein

an attribute as the lower-level element is structured by defining a descriptor and

a specific definition.

[Claim 5]

A data processing device for generating feature data representing characteristics of contents of input data, said processing device comprising:

means for generating feature data on the basis of, a descriptor becoming a unit element defining a describing method, which includes structure to have its attribute defined as a lower-level element and which may include a descriptor as the attribute.

[Claim 6]

The data processing device according to claim 5, wherein

the input data includes at least one of visual data or audio data included in video data.

[Claim 7]

The data processing device according to claim 5, wherein

the descriptor is generated to inherit functions including a capability of forming the structure from a higher-class descriptor.

[Claim 8]

The data processing device according to claim 5, wherein

an attribute as the lower-level element is structured by defining a descriptor and a specific definition.

[Claim 9]

The data processing device according to claim 5, wherein

the feature data is sent along with the input data to an external equipments.

[Claim 10]

A data processing device for utilizing feature data representing characteristics of contents of input data, said processing device comprising:

means for restoring features of contents of the input data by utilizing feature data generated on the basis of, a descriptor becoming a unit element defining a describing method, which includes structure to have its attribute defined as a lower-level element and which may include a descriptor as the attribute.

[Claim 11]

The data processing device according to claim 10, wherein

the input data includes at least one of visual data or audio data included in video data.

[Claim 12]

The data processing device according to claim 10, wherein

the descriptor is generated to inherit functions including a capability of forming the structure from a higher-class descriptor.

[Claim 13]

The data processing device according to claim 10, wherein

an attribute as the lower-level element is structured by defining a descriptor and a specific definition.

[Claim 14]

The data processing device according to claim 10, wherein

the restoring means restores features of the input data and generates a processed one of the input data.

[Claim 15]

The data processing device according to claim 10, wherein

the feature data is received along with the input data from an external equipments.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention]

The present invention relates to a data describing method for describing feature data which represents characteristics of contents of input data. The present invention also relates to a data processor for generating the feature data. Further the present invention relates to a data processor in which the feature data is utilized.

[0002]

[Prior Art]

It is often desired to search, for playback, a desired part of an interesting part or the like in a video application composed of a large amount of different video data, such as a television program recorded in video data, for example.

[0003]

In this manner, as a typical technique to extract a desired visual content, there

is a storyboard which is a panel formed from a sequence of images describing a main scene in an application. The storyboard is prepared by decomposing video data into so-called shots and displaying representative images of the respective shots. Most of the image extraction techniques of this sort are to automatically detect and extract shots from a video structure as disclosed in "G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, J. of Visual Communication and Image Representation 7: 28-4, 1996", for example.

[0004]

[Object of the Invention]

Meanwhile, for extracting the aforementioned video structure, the structure has had to be captured at various levels corresponding to contents of the video data. Therefore, processing for extracting such a video structure is very complex, so that an apparatus intended for extraction of video structures should have a high capability of processing and a large volume of computation resources. Thus, if required costs and labor are considered, it has been very difficult to implement such processing in the home.

[0005]

The present invention has been made in view of the above situation, and has an object to realize a highly advanced method for retrieving and browsing contents of video data, thereby to provide data describing method for describing data to generate so-called meta-data acquired particularly by automatic content analysis of various

video data in the home or the like. In addition, the present invention has an object to provide a data processor for generating such data. Further, the present invention has an object to provide a data processor to generate meta-data which provides easy and private access to the video-contents using such data.

[0006]

[Means for Achieving the Object]

A data describing method according to the present invention to attain the above object, is one for describing feature data which represents characteristics of contents of input data, and characterized in that a descriptor becoming a unit element which defines a describing method, includes structure to have its attribute defined as a lower-level element, and may include a descriptor as an attribute.

[0007]

Such data describing method according to the present invention, describes feature data structured by using a set of descriptors including structure.

[0008]

Also the data processor according to the present invention to attain the above object, is one for generating feature data which represents characteristics of the content of input data, and characterized by comprising generating means which generates feature data on the basis of, a descriptor becoming a unit element defining a describing method, which includes structure to have its attribute defined as a lower-level element and may include a descriptor as an attribute.

[0009]

The data processor prepared as described above, according to the present invention, generates feature data structured by using a set of descriptors which includes structure.

[0010]

Further, the data processor according to the present invention to attain the above object, is one for utilizing feature data representing characteristics of contents of input data, and is characterized by comprising restoring means which restores features of contents of input data by utilizing the feature data generated on the basis of a descriptor becoming a unit element defining a describing method, which includes structure to have its attribute defined as a lower-level element and may include a descriptor as an attribute.

[0011]

The data processor structured as described above, according to the present invention, restores features of contents of input data on the basis of feature data.

[0012]

[Embodiments]

An embodiment to which the present invention is applicable, will be described in detail below with reference to the accompanying drawings.

[0013]

An embodiment to which the present invention is applicable, is, for example,

a data transmitting-receiving processing system including a transmitter used in a broadcasting station or the like to supply multimedia-contents such as video data composed of visual and audio data, and a receiver to receive, for processing, multimedia-contents from a transmitter such as a digital video recorder, home server or the like. The data transmitting-receiving processing system is assumed herein to deal with video data as multimedia-contents. Before proceeding to the detailed description of the data transmitting-receiving processing system, video data as multimedia-contents, to which the present invention is applicable, will be described here first.

[0014]

Video data to which the present invention is applicable, is modeled as shown in FIG. 1, and is assumed to include structure classified to three hierarchical levels such as a frame, a segment, and a scene. That is, the video data is formed from a sequence of frames at the lowest level. Also the video data is formed from a sequence of consecutive segments as a level one step higher than the level of the frames. Further, the video data is formed by gathering together the segments on the basis of some meaningful relation, as a level one step higher than the level of the segments. In addition, the video data includes a furthermore higher level at which there is laid a topic not shown formed by grouping together related scenes, if necessary. Namely, the video data can be also modeled with the frames, the segments and the scenes shown in FIG. 1 replaced respectively with segments, scenes and a topic.

[0015]

The video data is a set of synchronized visual and audio streams. That is, the frames in the video data include visual frames each being a single still image, and audio frames representing audio information having generally been sampled for a time as short as tens to hundreds of milliseconds.

[0016]

The segment include visual segments and audio segments, the visual segments being comprised of a sequence of visual frames having consecutively been picked up by a single camera and generally called " shot". This segment is a fundamental unit of video structure. In the segment, the audio segment can be defined in many different manners, for example, a way described below can be considered. For example, audio segments are formed bounded by periods of silence in video data detected by the well-known method, as the case may be. Also, in some cases, an audio segment is formed by a sequence of audio frames classified in several categories such as voice, music, void, silence, etc., as disclosed in "D. Kimber and L. Wilcox, Acoustic Segmentation for Audio Browsers, Xerox Parc Technical Report", for example. Further, in other cases, the audio segments are determined by using an audio cut detection for detecting a large variation of a certain feature between two successive audio frames, as disclosed in " S. Pfeiffer, S. Fischer and E. Wolfgang, Automatic Audio Content Analysis, Proceeding of ACM Multimedia 96, Nov. 1996, pp21-30".

[0017]

Further, to gather together, for grouping, the contents of video data at a higher level having semantics, the scene is made up of a meaningful group formed from segments which captured visual frames or audio frames by detecting visual segments (shots) or audio segments, by using feature information (feature) representing features of segments such as, for example, an amount of perceptual activity in the segments.

[0018]

Furthermore, the topic represents a sequence of semantically related scenes.

[0019]

A data transmitting-receiving processing system, shown in FIG. 2 as an embodiment to which the present invention is applied, includes a transmitter 10 to supply a variety of data such as the aforementioned video data as multimedia-contents, and a receiver 20 to receive the various data such as video data from the transmitter 10.

[0020]

As shown in FIG. 2, the transmitter 10 includes a memory part 11 to store the aforementioned video data, a feature data generator 12 to generate feature data which represents characteristics of contents of video data supplied from the memory part 11 and describes video structure or the like, an encoder 13 to encode multimedia-contents supplied from the memory part 11, and an encoder 14 to encode feature data supplied from the feature data generator 12.

[0021]

As shown in FIG. 2, the receiver 20 includes a decoder 21 to decode encoded multimedia-contents supplied from the transmitter 10, a decoder 22 to decode encoded feature data supplied from the transmitter 10, and a retrieval engine 23 as a restoring means to analyze video structure based on feature data and provide desired contents to the user.

[0022]

In the data transmitting-receiving processing system, the transmitter 10 generates feature data describing video structure of video data held in the memory part 11 or the like, and sends it along with the video data to the receiver 20 via a transmission path not shown. Based on received feature data, the receiver 20 restores the video structure of the received video data and generates a storyboard as data which makes it possible to browse a sequence of frames representative of each of the aforementioned scenes for example.

[0023]

Thus, in the data transmitting-receiving processing system, for example, when the user supplies a request for viewing a desired content of the video data to the receiver 20 via the retrieval engine 23, the receiver 20 can analyze the video structure of the restored video data by the retrieval engine 23 and provide the desired content to the user. Thus, the user can easily and privately access to the video data and make an efficient browsing.

[0024]

A described content of feature data for use to extract higher-level video structure by using lower-level feature extracted from the content of video data, will further be described in detail below.

[0025]

A describing method for describing a feature of the visual and audio contents of video data is herein defined as description scheme. The description scheme is composed of a video core description scheme (will be referred to as "VCS" hereinafter) which is used to describe basically features of the element contents of both visual and audio of video data, and a video browsing description scheme (will be referred to as "VBS" hereinafter) which is intended to extend VCS by a descriptor being a descriptor which permits to visualize the structure represented by the VCS, especially for browsing of the video data.

[0026]

VCS can be used to capture a variety of video structure in video data, and it permits to capture structures of between visual segments and audio segments in the video data and its correlation. In addition, VCS captures side aspects of three different structures by means of superposing layers in a hierarchical manner and integrates in a single logical describing method, that three different structures including a classification (similarity) structure formed by combining segments similar to each other in the video data, temporal segmentation structure being a set of temporally consecutive segments, and a logical structure. Further, VCS describes the video data

at both of a lower-level in the video data and a semantic structure by using a variety of ideational levels. Furthermore, VCS enables to access to information on video data at multi-layered detailed levels.

[0027]

On the other hand, VBS constructs video data logically, permitting the user to understand the content of the video data and find points of interest to him or her. Also, VBS enables to retrieve not only lower-levels such as frames but also high-levels such as scenes by means of comparison on the basis of the similarity.

[0028]

The description scheme of this sort is a set of descriptors being its descriptor.

[0029]

The descriptor can have a structure by defining its attribute and which is formed by defining other descriptors including itself and special value as lower-level element. The descriptor has a description format shown in FIG. 3, for example. That is, the description format of the descriptor is composed of "Descriptor" for denoting the name of a descriptor and the descriptor code, "Superclass" for denoting a descriptor in a higher class than the descriptor, "Feature" for denoting a descriptor as lower-level element or a sort of numerical value, "Feature Type" for denoting a descriptor as lower-level element or a type of numerical value, and "Example Value" for denoting a descriptor as lower-level element or an example of numerical value.

[0030]

The higher-class descriptor denoted as the Superclass refers to an original descriptor, from which a certain descriptor is inherited in a so-called object-oriented expression. That is, the descriptor, herein, is described by the object-oriented expression and inherits functions from an original descriptor. Thus, the descriptors can more simply and easily be extended to ones having higher functions so that feature data described using the descriptor can be structured with a more simple structure having less information. The relation of inheritance between the descriptors herein is assumed to be shown in FIG. 4.

[0031]

As shown in FIG. 4, the descriptors belong to any of VCS or VBS, and form a single system.

[0032]

More specifically, the descriptors belonging to VCS include Audio-Visual-Element (Audio Visual Element; will be referred to as AV element or AVElement hereafter), Audio-Visual-Segment (Audio Visual; Segment will be referred to as AV segment or AVSegment hereafter), Audio-Visual-Abstract (Audio Visual Abstract; will be referred to as AV abstract or AVAbstract hereafter), Audio-Visual-Prototype (Audio Visual Prototype; will be referred to as AV prototype or AVPrototype hereafter), Audio-Visual-Hierarchy (Audio Visual Hierarchy; will be referred to as AV Hierarchy or AVHierarchy hereafter), Cluster, Composite Segment, Audio-Visual-Graph (Audio Visual Graph; will be referred to as AV Graph or

AVGraph hereafter), Audio-Visual-Graph-Edge (Audio Visual Graph Edge; will be referred to as AV Graph Edge or AVGraphEdge hereafter), Signature, Temporal Signature, Activity Level, Weight Vector, Color, Color Space, Concept, and Color Signature. XML and RDF are also included as reference information.

[0033]

On the other hand, the descriptors belonging to VBS include Synopsis, Audio Frame (will be referred to as audio frame), Visual Frame (will be referred to as visual frame), Browsing-Node, Shot, Scene, Topic, Program, and Highlight.

[0034]

In FIG. 4, a descriptor with △ is a higher-class one in the relation of inheritance between descriptors, and a descriptor connected to a descriptor with ◇ is one that expresses its attribute. That is, in the relation of inheritance between descriptors, AVSegment, AVAbstract, AVHierarchy, Cluster and AVGraph inherit from AVElement. Also, in the relation of inheritance between descriptors, the descriptor which expresses an attribute of AVGraph is AVGraphEdge. Also, for other descriptors, descriptors from which the other descriptors inherit and those expressing attributes of the other descriptors can be specified as like the above.

[0035]

These descriptors will be described one after another below. It should be noted that in the following description, it is assumed that video data consists of one visual stream and one audio stream. However, it should also be noted that the description

schemes could also be used for ordinary video data consisting of an arbitrary number of visual and audio streams.

[0036]

First, the AVElement descriptor as a descriptor belonging to VCS will be described. The AVElement is introduced as an abstract descriptor representing a logical unit intended for use to analyze the content of a visual audio stream. The AVElement descriptor of this sort is undefined intentionally so as to include together an arbitrary grouping based on a certain classification method, such as basic analysis unit like a visual frame, higher-level structure unit like shot and scene, all frames in which a certain actor or actress appears.

[0037]

The description format for the AVElement descriptor of this sort is as shown in FIG. 5. Note that, AVE in a column of a descriptor expressing the name of a descriptor and a descriptor code shown in FIG. 5 expresses a descriptor code. The AVElement descriptor defines an abstract element intended for description of the content of a visual-audio stream. In the description scheme, all the other nodes describing some side aspects of the content of a visual-audio stream inherit the AVElement descriptor. AVElement descriptor may designate an actual part of video data, or also an abstract or analytic composition formed to describe the content of video data.

[0038]

In the description scheme, a node of the video data always uses a type of this sort in the case that one node needs to refer to another describing element. Thus, a different content analysis tools are mixed mutually freely. Also in this means, abstraction is going up from lower-level to high-level. In the lowest-level, AVElement describes an element which is close to live media, and a feature of contents-level in a higher-level. In both of the cases, the same set of the tool for description can be used in hierarchy manner and linked mutually.

[0039]

In the same manner, a feature value (feature) can be added freely not only to its original value but also to a structural obtained analytically. For example, it can be added not only to a scene where a color histogram is single, but also to the AVElement expresses a group of shots.

[0040]

Next, the AVSegment descriptor will be described below. The most important AVElement in the description scheme is an AVSegment descriptor. The AVSegment designated a cut time interval of a video stream including at least one or more visual audio streams. A set of the streams may include visual, audio or the both type of streams. The time interval may be designated by a common time added to video data, or may be specified by or common frame numbers or the like assigned to video data. An example of the AVSegment is a content of a series of frames which expresses one of the video frame or a perfect camera shot.

[0041]

The description format for the AVSegment descriptor is as shown in FIG. 6., and the AVSegment includes the AVElement as the Superclass. Also, the AVSegment has, as features denoting other descriptors or sorts of numerical value as lower-level element elements, Start Point whose type is Timecode and which denotes the start point of the segment in the video data, End Point whose type is Timecode and which denotes the end point of the segment in the video data, Streams whose type is a designated set of "streamref" and which denotes a set of streams designated by the segment in the video, Abstract whose type is a set of AVAbstract, which denotes more than zero feature denoting a summed visual-audio content of the segment and whose Example Value is a key frame, and Concept whose type is a set of Concepts, which denotes more than zero commentary which gives a visual audio semantic content of the segment and whose Example Value is a RDF document which describes a content of the segment. Note that, an AVS in a column of the descriptor, which denotes the name of a descriptor and its descriptor code shown in FIG. 6 represents a descriptor code.

[0042]

The AVSegment itself does not have its abstract level, domain and semantics in it. The AVSegment can be used to denote both of lower-level unit such as single visual frame and high-level unit such as commercial in a TV program.

[0043]

The minimum AVSegments in the visual stream or audio stream, will be referred to as a "visual audio frame". Namely, in the visual stream, the visual frame refers to a single still image as having previously been described. On the other hand, in the audio stream, the audio frame refers to normally a short sequence of audio samples, not to a single sample, to make it possible to be analyzed. In the description scheme, the visual audio frames are dealt with as the simplest node of the Audio segment, since the interest is in the segment more than frame. Other examples of the AVSegment are shots, scenes and the like.

[0044]

In the description scheme, a Timecode descriptor is utilized as a placeholder. The only requirement is that the Timecode descriptor should be a numerical scale enabling to measure time duration and including a common time scale to all streams in video data. More specifically, the time should be measured in seconds of a time from the beginning of video data, for example. In the same manner as described above, the Streamref descriptor is a placeholder for a descriptor to make reference to a single stream in a multi-stream-multi-media text.

[0045]

Next, the AVAbstract descriptor will be described below. The description format for this AVAbstract descriptor is as shown in FIG. 7, and has the AVElement descriptor as the Superclass. Note that, an ABST in a column of the descriptor, which denotes the name of a descriptor and its descriptor code shown in FIG. 7, represents

a descriptor code.

[0046]

The AVAbstract descriptor expresses an abstract of the content of the AVElement. For example, the AVAbstract descriptor selects a lower-level element being able to be representative (when this element denotes a scene, for example, the lower-level element will denote a shot), and will denote it as an abstract, as the case can be. Ideally, AVElement used for the abstract includes two properties. Namely, the first property represents an important or appropriate content of the AVElement. The second property represents a feature of the AVElement and also represents a deference of the AVElement from the other AVElements.

[0047]

The abstract can be utilized for substitution of a real content of the video data. In the description scheme, the most important use of this AVAbstract descriptor is to retrieve similar AVElements just by comparing the abstracts. That is, this abstract serves as a key for detection, comparison, retrieval, and the like of the AVElements.

[0048]

Next, the AVPrototype descriptor will be described below. The description format for this AVPrototype is as shown in FIG. 8. This AVPrototype has the AVAbstract descriptor as the Superclass. The AVPrototype descriptor, as a feature which denotes a descriptor or sorts of numerical value of lower-level element, contains Kind whose type is String and which represents a type of AVElement represented by

the AVPrototypre descriptor as well as representation of a feature of the AVPrototype descriptor, and whose sample value is Shot, Scene or VisualFrame, Weights whose type is Weight Vector and which indicates n-dimensional vector representing a weighting of the association or importance of a feature regarding each feature of the prototypes, and, as necessary, a plurality of features 1, ..., n(Feature1, ..., Featuren) whose type are descriptor 1, ..., n (Descriptor1, ..., Descriptor n) and which indicate feature examples of descriptor value from the first to the n whose type is an arbitrary descriptor. Note that, an AVP in a column of the descriptor, which denotes the name of a descriptor and its descriptor code shown in FIG. 8 represents a descriptor code.

[0049]

The AVPrototype is an outlining expression of the AVElement. Particularly, in the description scheme, this AVPrototype can describe a prototype-like instance of an AVElement of a certain type as a set of descriptor values. After that, an association or importance of each feature is defined specifically by the weights. In other words, a certain prototype element intends for use of Collating with a very simple instance, and describes a specified sort of AVElement with the deserved Feature Value.

[0050]

Generally, a prototype is structured by analysis and utilized in case that the specific representative AVElement needs to be positioned against a certain group of AVElements. For example, to determine how much a certain AVElement is similar to a certain set of AVElements, it is possible to compare the similarities of the

prototype of the element and the prototype of the group.

[0051]

Note that, all images included in a single shot for example will be described as shown in FIG. 9, it is desired to define a prototype image for this group of images. In this case, firstly, different feature value such as ranges of color and texture for an image is obtained in place of a real image. Next, a typical value such as a mean-value for example is determined for each feature. This value is a feature value in the prototype. And then, a certain feature determines how much relation exists for the set in the shot. For example, in the shot, although color is very important, texture does not have so much association as the case may be. A difference of association or importance refer to each feature of this sort is represented by weights allotted to each feature, and the magnitude of the value indicates association to the prototype for the feature value.

[0052]

In this manner, the AVPrototype descriptor describes a frame representative of segments used in structure analysis of video data such as shots and scenes. For example, as will be further described later, using a frame representative of the shot described by AVPrototype, extracting feature values for each of representative frames as feature values for the shots, comparing the values with each other, detecting similar shots lying close to each other on the time basis and merging repeatedly, a scene are generated a scene finally. Note that, in the description scheme, a frame representative

for a user to browse video data is not Avprototype descriptor but a Cluster descriptor which will be described later.

[0053]

Next, the Hierarchy descriptor will be described below. Grouping together visual segments and audio segments in the video data that are similar to each other is a basic requirement for structuring of video data for retrieval the content or to browsing. In the description scheme, using a hierarchical structure based on a tree, it is proposed to group sets of visual segments and audio segments at different detailed levels. The hierarchical classification provides the following features. Namely, the first feature is to enable to access different detailed levels, and the second feature is to provide a content abstract. The third feature is to provide a hierarchical classification method and the fourth feature is to enable to capture an approximate semantic classification in high-level, not perceptual similarity in lower-level.

[0054]

Each node in the hierarchy denotes a certain set of AVElements. That is, among the each node in the hierarchy, some are belonged to the node directly and some are included repeatedly to at least one or more subsidiary node of the node. When a descriptor is represented with a n-dimensional vector, a centroid of the descriptor value can be a preferable abstract for a feature that a plurality of members of the Hierarchy or representative members is abstracted by the each classification in the hierarchy.

[0055]

The description format for the AVHierarchy descriptor is as shown in FIG. 10. This descriptor includes the AVElement descriptor being the Superclass. Also, the Hierarchy has, as features denoting a descriptor or a sort of numerical value of lower-level element, Root whose type is Cluster and which denotes a start point of the hierarchy, and Kind whose type is String and which denotes the kind of the hierarchy. Note that, an HIER in a column of the descriptor which denotes the name of a descriptor and its descriptor code shown in FIG. 10 represents a descriptor code.

[0056]

The AVHierarchy descriptor of this sort denotes a result of grouping a plurality of AVElements related with or similar to each other into one hierarchical structure.

[0057]

Next, the Cluster descriptor will be described below. The description format for this Cluster descriptor is as shown in FIG. 11. This Cluster descriptor has the AVElement being the Superclass. Also, the Cluster has, as features denoting a descriptor or a sort of numerical value of lower-level element, Members whose type is a set of AVElements and which denotes a sets of AVElements included directly in the group, and Struct whose type is AVAbstract and which denotes value expressing a member representative of the set of AVElements extracted from a set of segments included in the Cluster.

[0058]

The Cluster descriptor groups together AVElements related to or similar to each other to form one class. This definition is recurring, and some clusters may include other clusters as members. In this case, the result will be a hierarchical classification structure.

[0059]

Using the Cluster descriptor, segments clustering can be carried out based on the similarity between the segments as shown in FIG. 12, and the result of clustering can be described. More specifically, use of the Cluster descriptor enables to cluster shots based on their similarity as will further be described later.

[0060]

A definition of the Cluster descriptor hardly gives a limitation to members included in the hierarchical classification structure and classification nodes. Therefore, the following limitation will be added to a classification tree for many cases.

[0061]

Namely, the first limitation is to limit the number of elements of subsidiary nodes. That is, the number of the subsidiary nodes permitted in each Cluster descriptor may be limited. For example, when the number of elements is limited to two, binary tree structure can be formed. Also, when each node has fixed element number, a structure based on a binary tree, fourth scale tree or eighth scale tree can be prescribed.

[0062]

Also, the second limitation is to limit partition. That is, when some visual segments and/or audio segments are partitioned to Clusters, elements of the segments cannot be elements of two Clusters by duplication. Also, all of the elements must be elements of one Cluster.

[0063]

Further, the third limitation is to limit levels. According to the definition described above, any height of trees can be formed. For example, it is not necessary that the tree must be over first level. However, classification method has this sort of limitation may possibly exist.

[0064]

As a method for forming the similarity tree of this sort, a plurality of methods is well known. For example, agglomerative hierarchical clustering methods or the like are so.

[0065]

Meanwhile, by seeing hierarchical classification from slightly different angles, the hierarchical classification can be interpreted as a hierarchical quantization method for obtaining descriptor value. Particularly, quantization value is selected to the descriptor by allotting arbitrary descriptor value to the most similar classification group. And then, AVPrototype value refers to the group is used as quantization value.

[0066]

For example, suppose vector quantification method represented in n-dimensional vector space for descriptors. A vector for use of quantification of a set with some value is generally found by allotting feature space to a set of ranges using some of generalized clustering algorithm such as Lloyd algorithm, Kohonen Self-Organizing Maps, or the like, so that some code book for quantization is formed. When one Cluster is allotted to an entry of each code book and the code book value is used for prototype value, VQ code book can be described using hierarchical classification.

[0067]

Next, the Composite Segment descriptor will be described below. The hierarchical classification captures similarity between groups of AVSegments independently of the temporal sequence of the segments between segments. A concept of the Composite Segment, which captures a temporal hierarchical structure of video data, is introduced. The Composite represents segments formed from a sequence of elements of AVElements, as shown in FIG. 13, for example.

[0068]

The description format for the Composite Segment descriptor is as shown in FIG. 14. This descriptor has the AVSegment being the Superclass. The Composite Segment has, as a descriptor or a sort of numerical value of lower-level element, Components whose type is a set of AVSegments. Note that, SN in a column of the descriptor, which denotes the name of a descriptor and its descriptor code shown in

FIG. 14 represents a descriptor code. Further, since the Composite Segment descriptor is a special AVSegment descriptor, the definition of the Components makes it possible to generate segmentation repeatedly.

[0069]

The Composite Segment descriptor describes how a sequence of AVSegments is segmented into sub sequences similar to or related with each other. Using this Composite Segment descriptor, it is possible to describe the structure of video data as a tree structure. More specifically, use of the Composite Segment descriptor describes as a tree structure a result of clustering of similar segments by grouping, and also describes the structure of video data as a tree structure by further clustering the scene.

[0070]

Next, the AVGraph descriptor and AVGraphEdge descriptor will be described below. To capture the logical structure of video data, a concept of the AVGraph is introduced. A media structure graph is an oriented-graph, in which each apex describes an AVElement such as an audio segment and each edge describes a transition from a certain element to the other elements. Two specific apexes are specified as a starting apex and an ending apex.

[0071]

A fundamental concept of the AVGraph descriptor is to enable to apply some simple tool for describing formal language to description of a logical structure of video data. For example, as is the case with the formal language is based on a concept of a

sequence of symbols taken from the alphabet, a stream of the video data can be described as a sequence of the AVElements. AVGraph can be considered as a one variation of the finite state automaton whose state has been replaced with AVElement. Two different media structure graphs for a sequence of shots which forms a dialogue scene as shown in FIG. 15 (A), are shown in FIG. 15 (B) and 15 (C). An instance shown in FIG. 15 (B) describes a simple straight structure, and an instance shown in FIG. 15 (C) describes a change structure in the high-level, which has a feature being shots changes deepening on a speaker.

[0072]

The description format for the AVGraph descriptor is as shown in FIG. 16. This descriptor has the AVElement as the Superclass. Also, the AVGraph descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Edges whose type is a set of AVGrahEdges and which denotes a set of edges in the graph, and Kind whose type is String and which denotes the kind of the graph. Note that, AVG in a column of the descriptor which denotes the name of a descriptor and its descriptor code shown in FIG. 6., represents a descriptor code.

[0073]

The AVGraph descriptor describes connection between a pair of AVElements to a graph including "Edges" added with labels denoting different sort of association between a pair of AVElements. AVGraph is a graph added with a label which has a node indicating an AVElement, and an edge added with a label which links each node.

[0074]

The AVGraph descriptor makes it possible to describe a repeated structure of video data. More specifically, by using the AVGraph descriptor, the content of detected scenes is described as a repeated structure of the scenes, and used for analyzing the entire structure of the video data.

[0075]

On the other hand, the description format for the AVGraphEdge descriptor is as shown in FIG. 17. The AVGrapEdge descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Source whose type is AVElement and which denotes a source element of the edge, Target whose type is AVElement and which denotes a target element of the edge, and Label whose type is String and which indicates the kind of the relation between two elements connected by the edge of the two elements. Note that, AVEDGE in a column of the descriptor which denotes the name of a descriptor and its descriptor code shown in FIG. 17, represents a descriptor code.

[0076]

The AVGraphEdge descriptor of this sort indicates the directed connection between two AVElements.

[0077]

All the AVGraphEdges belong to a certain AVGraph. Also, the same AVElements belong to different graphs, respectively, as the case may be. Further, the

same AVElements of two can be connected to each other in different AVGraphs, respectively.

[0078]

In the definition of the AVGraph, no special constraints are imposed upon the type of AVElements to which the node can make a reference. Particularly the node can be linked to the AVPrototype. In this case, the graph will be referred to as "AVPrototype Graph", and indicates the prototypical instance as a prototype of a structure of streams of the video data. Such a structure can be used to represent a structure in a TV news program for example.

[0079]

Also, the AVGraph can form a hierarchy of graph in which one graph is actually formed to a nested structure as like another graph. Thereby the AVGraph enables to capture representing structures at the multiple levels.

[0080]

Next, the Signature descriptor will be described below. The description format for the Signature descriptor is as shown in FIG. 18. The Signature descriptor has the AVAbstract descriptor as the Superclass. Also, the Signature descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Weights whose type is WeightVector[n] and which denotes n-dimensional vector of weight, Members whose type is an AVElement[n], and which denotes AVElements to each of weighting and weights and AVElment correspond as one to one, and Kind

whose type is String and which denotes kind of the Signature and only same kind of Signatures enable to compare with each other. Note that, SIG in a column of the descriptor which denotes the name of a descriptor and its descriptor code shown in FIG. 18, represents a descriptor code.

[0081]

The Signature descriptor represents the static contents of the AVElements by combinations of a plurality of representative AVElements and Weights of their respective elements.

[0082]

One Signature can be represented as a generalized histogram. AVElement to which the Signature objects, can be expressed as a distribution state of category elements, i.e., a distribution of AVElements descriptor of Signature members. Weighting indicates a part of distribution allotted to the specific class, or a probability classification vector. In this case, each weight represents probability that descriptor value is a member of the class corresponding to the value.

[0083]

A first example of the Signature is Shot Key Frames. The Shot Key Frames is a set of frames representative of shots extracted from the contents. The weight of each key frame denotes a ratio of frames matching the key frame (being most similar to the key frame).

[0084]

A second example of the Signature is Program Signature. The Program Signature is a set of shots indicating definitely the features of a certain program. For a news program, for example, a shot in which the opening logo mark appears, shot of the newscaster, shot of the last logo mark, or the like can be used as best representing the features of the news program.

[0085]

As having been described above, the Signature descriptor designates a method or parameter for the feature extraction of video data. More specifically, by using the Signature descriptor, the method will be described, when the best feature extracting method in correspondence to the kind of contents, as will be described later.

[0086]

Next, the Temporal Signature descriptor will be described below. The description format for the Temporal Signature descriptor is as shown in FIG. 19. As seen, the Temporal Signature descriptor has the AVAbstract as the Superclass. The Temporal Signature descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Time whose type is WeightVector and which denotes a relative time duration of each section of nodes, Members whose type is a sequence of AVElements(seq of AVElement) which is a sequence of visual and audio representing features of nodes, and Kind whose type is String and which denotes a kind of the Signature, and the same kind of Signatures enable to compare to each other. Note that, TS in a column of the descriptor which denotes the name of a

descriptor and its descriptor code shown in FIG. 19, represents a descriptor code.

[0087]

Such a Temporal Signature descriptor represents the dynamic content of the object AVElement the Temporal Signature descriptor by a sequence of a plurality of representative AVElements being members of the Temporal Signature descriptor. Each member of visual and audio sequences represents one state of the object AVElement. Weight is provided to each visual and audio sequence and denotes the relative length of the state corresponding to the sequence.

[0088]

The number of necessary visual and audio sequence for the object AVElement should depend on both the extent of change of the AVElement itself and the detail of a content. In the least case, when the change is very small, the AVElement can be dealt as one sequence of visual and audio. On the other hand, in the most common case, a sequence of visual and audio is made of all of sub elements of the object AVElement, as the case may be. This is the case that all frames of the shot turn into a sequence of visual and audio.

[0089]

A first example of the Temporal Signature is Shot Temporal Signature. It is a sequence of key frames denoting the important time in a motion of the shot, and represents characteristics of a shot. For example, a zooming scene from the shot to the player is the one.

[0090]

A second example of the Temporal Signature is Television Program Temporal Signature. This is a set of key shots extracted from a program and represents characteristics of the program. In case of news program, for example, a shot of a representative logo mark, shot of the newscaster, shot of sports, weather forecast, or the like can be used as best characterizing the news program.

[0091]

Both of the aforementioned Signature and this Temporal Signature are described by sequences of visual and audio to which AVElement is weighted. However, different from the Signature, the temporal Signature has temporal order. The Temporal Signature describes a sequence in the state that the AVElement passes based on the time elapses. The weighting factor is an index indicating how much (relative) time required in each class. Namely, the order of classes is in correspondence to the temporal order.

[0092]

The Temporal Signature descriptor of this sort designates a temporary feature extracting method or a parameter for feature extracting of video data. More specifically, Temporal Signature descriptor is utilized for the process to search the best solution by repeated trial and error in the case that the best feature extracting method is selected in correspondence to the kind of the contents, as will be described later.

[0093]

Next, as an Audio Visual descriptor, an Activity Level descriptor, Weight descriptor, Weightvector descriptor, Color descriptor, ColorSpace descriptor, Concept descriptor, and ColorSignature descriptor will be described below. The descriptions relates to these descriptors are applied to all of AVElements, and recorded as so in columns to which the descriptions are applied in the description format.

[0094]

The description format for the Activity Level descriptor is as shown in FIG. 20. As seen, this descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, ActivityLevel whose type is Real and which is real number of 0 or over, and 1 or less; 0 indicates that there is no activity while 1 indicates the activity is maximum. Note that, ACT in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 20, represents a descriptor code.

[0095]

The Activity Level descriptor denotes an extent of the Audio Visual Activity level. The method for generating the Activity Level descriptor depends upon necessity of an application.

[0096]

For example, visual segments will be considered here. The activity level in the visual sequence can be measured by various methods, such that the activity level can be measured by a mean distance between adjacent frames in a sequence, entire

momentum taking place in the sequence, or the like.

[0097]

With the Activity Level descriptor, visual and audio features are described as will further be described later, and used as a parameter for measurement of how much the content of video data changes.

[0098]

The description format for the Weight descriptor is as shown in FIG. 21. As seen, this descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Weighting whose type is Real and which is real number between 0 to 1, indicating the amount of weight added to the AVElement. Note that, WGT in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 21, represents a descriptor code.

[0099]

The Weight descriptor denotes a relative weighting of the AVElement. A meaning of weighting is determined based on the only content in which the weighting is used.

[0100]

The description format for the WeightVector descriptor is as shown in FIG. 22. This descriptor has, as features denoting a descriptor or a sort of numerical value attribute of lower-level element, Weights whose type is Real [n] and which is a n-dimensional vector of the Weight, and Normalized whose type is Boolean and which

is a flag indicating whether or not a value included in the Weights is standardized. Note that, WV in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 22, represents a descriptor code.

[0101]

The WeightVector descriptor denotes an n-dimensional vector of weighting applied to the AVElements. With this WeightVector descriptor, a weighted value of a parameter at feature extraction or a parameter obtained as a result of feature extraction from video data, is designated or described. More specifically, the WeightVector descriptor designates a extent of contribution for judgment with respect to each parameter in the process of judging similarity, to every parameters indicating a plurality of similarities, as will be further described later.

[0102]

Further, the description format for the Color descriptor is as shown in FIG. 23. The Color has, as features denoting a descriptor or a sort of numerical value of lower-level element, Color whose type is Real [3], which indicates a color element value represented in a range of [0. 0, 1. 0]. Note that, COL in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 23, represents a descriptor code. Also, color is modeled as three-dimensional vector whose element is real number, and each element takes value from 0 to 1.

[0103]

The Color descriptor describes a color with three different real numbers. Each

value corresponds three axes in ColorSpace, respectively.

[0104]

The description format for the ColorSpace descriptor is as shown in FIG. 24. The ColorSpace has, as features denoting a descriptor or a sort of numerical value of lower-level element, ColorSpace whose types is String and which indicates a ColorSpace name. Note that, CS in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 24, represents a descriptor code.

[0105]

Color can be encoded with various and different ColorSpaces. The ColorSpace descriptor determines how the meaning of the Color descriptor should be interpreted.

[0106]

Further, the description format for the Concept descriptor is as shown in FIG. 25. The Concept has, as features denoting a descriptor or a sort of numerical value of low-element, Concept whose type is RDF node, XML node or the like, and which indicates a linkage to the concept of the AVElement, such as linkage to the description based on the RDF node or XML node. Note that, CPT in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 25, represents a descriptor code.

[0107]

This Concept descriptor links the AVElement to expression which describes the

conceptual contents. In the description scheme, it is not designated what kind of formalism should be utilized for describing the AVElement conceptually.

[0108]

Further more, the description format for the ColorSignature descriptor is as shown in FIG. 26. The ColorSignature has, as features denoting a descriptor or a sort of numerical value of lower-level element, ColorSpace whose type is ColorSpace, which indicates a ColorSpace in which the histogram is calculated, and whose instance value is RGB, Colors whose type is Color [n] and which indicates n-dimensional vector being a set of colors with which the similarity is defined, and Weights whose type is WeightVector and which indicates n-dimensional vector of Weights representing image described with colors of corresponding Colors type. Note that, CHT in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 26, represents a descriptor code.

[0109]

The ColorSignature descriptor denotes a generalized color histogram. Conceptually, the histogram can be regarded as classification of a color into some categories and a frequency of appearance in the respective category of pixels in an image. The ColorSignature enables to generalize this method and set an arbitrary bin (minimum divisional area in a ColorSpace). Note that, each bin is represented by a representative color in the space.

[0110]

Assume here a certain color image for example. When it is desired to describe colors used in the image, first, colors are divided into areas. Each area will be classification items. Next, relative ratio of image pixels which are divided into the respective color area, is determined. This will be feature Weightings.

[0111]

As a descriptor belonging to VBS, the Synopsis-descriptor will be described below. A descriptor which provides a audio visual summary for browsing video data to the user will be described. The Synopsis is intended to describe some AVElements presented in a visual or audio manner or in both to the audience so as to summarize a part of the content of a certain video data.

[0112]

The description format for the Synopsis descriptor is as shown in FIG. 27. The Synopsis descriptor has the AVElement as the Superclass. The Synopsis has, as features denoting a descriptor or a sort of numerical value of lower-level element, Kind whose type is String, which indicates the kind of the Synopsis, and whose example value is a SLIDE SHOW, Sequence whose type is a list of AVElement and which indicates a list of AVElements forming the Synopsis, and IsOrdered whose type is Boolean and which is a flag indicating whether the order of the elements in the list is important or not. Note that, SYN in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 27, represents a descriptor code.

[0113]

The Synopsis descriptor denotes a summary of dynamic occurrence described in the AVElement. This denotes an order of the AVElement to be displayed when presenting the outline to the user.

[0114]

Apparently, the AVAbstract and Synopsis seem to present the same information. However, there is a large difference. Namely, the Synopsis is to abstract the AVElements for the user while the AVAbstract is to abstract for retrieving or indexing to AV.

[0115]

First example of the supposed Synopsis is Key frames. Items of the Synopsis are the ordered visual for describing action in the AVElements.

[0116]

A second example is Digest. Items in visual and audio are AVSegments. For example, a sequence of clips obtained from the video data.

[0117]

With the Synopsis descriptor, it becomes possible to grasp the content of the video data as the outline. More specifically, the Synopsis descriptor represents a detected scene by a representative frame detected by sampling of dynamic feature, and displays it as a storyboard, as will be further described later.

[0118]

Next, audio frame descriptor, visual frame descriptor, browsing node descriptor, shot descriptor, scene descriptor, topic descriptor, program descriptor, and highlight descriptor will be described as structure descriptors. As having previously been described, video data are constructed as a three-storied hierarchical structure. Namely, they are the segment, the scene and a topic herein.

[0119]

The description format for the audio frame descriptor is as shown in FIG. 28. The audio frame descriptor has the AVSegment as the Superclass. Note that, AF in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 28, denotes a descriptor code.

[0120]

The audio frame descriptor denotes an AVSegment corresponding to a single audio frame.

[0121]

On the hand, the description format for the visual frame descriptor is as shown in FIG. 29. The visual frame descriptor has the AVSegment as the Superclass. Also, the visual frame descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Color whose type is ColorSignature and which indicates the color characteristics of the frame. Note that, VF in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 29, represents a descriptor code.

[0122]

The visual frame descriptor is a special case of the AVSegment and denotes a single visual frame

[0123]

Further, the description format for the browsing-node-descriptor is as shown in FIG. 30. The browsing-node-descriptor has the Composite Segment descriptor as the Superclass. The browsing node descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Index whose type is a set of AVHierarchy and which indicates a set of hierarchical classification used to index the content of a certain AVSegment using the underlying sub elements and Sample Value is a classification of video data by way of example, Synopsis whose type is Synopsis and which indicates a outline of a certain part of the video data and Sample Value is a sequence of key frames by way of example, and Structure whose type is AVGraph (arbitrary) and which indicates a structure of the certain part of the video data. Note that BRO in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 30, denotes a descriptor code.

[0124]

The browsing node descriptor is a common abstract class for units of various video structures used in browsing.

[0125]

This browsing node descriptor can be used to describe the content of video data

in detail. Also, the browsing node descriptor can be used to the semantic content designated by RDF node as well as Synopsis descriptor and Activitylevel descriptor.

[0126]

Further more, the description format for the shot descriptor is as shown in FIG. 31. The shot descriptor includes the browsing node descriptor as the Superclass. Also, the shot descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Activity whose type is ActivityLevel,which indicates the activity level of the shot and Sample Value is 0.5 for example, Abstract whose type is FromAVSegment,Synopsis whose type is FromBrowsingNode and which indicates the synopsis-type abstract composed of a sequence of audio and/or visual frames, and Index whose type is FromBrowsingNode and which indicates an index to the shot composed of classification of audio and/or visual frames being arbitrary but structuring the shot, and Structure whose type is FromBrowsingNode. Note that SHT in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 31, represents a descriptor code. Also, the Abstract enables to carry out the following abstraction to the shot. A first abstraction is the TemporalSignature whose elements are audio frame, visual frame and AVPrototype, and a second abstraction is the Signature whose elements are audio frame and visual frame.

[0127]

The shot descriptor denotes a lowest level of the audio-visual unit, namely, a

shot.

[0128]

The shot descriptor can be used to describe the content of the shot of the video data specifically, namely, to inherit the Browsing Node descriptor for describing. More specifically, the shot descriptor can be used to designate a shot, based on which the clustering is carried out, for detecting scenes, as will be further described later.

[0129]

Next, the description format for the scene descriptor is as shown in FIG. 32. The scene descriptor includes the browsing node as the Superclass. The scene descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Kind whose type is String, which indicates a kind of a scene and whose Sample Value is DIALOG, COMMERCIAL, ACTION or ANCHOR, Abstract whose type is FromAVSegment, Synopsis whose type is FromBrowsingNode and which indicates an possible outline to the scene constructed by the synopsis-type data composed of a sequence of shots, Index whose type is FromBrowsingNode and which indicates an index to the scene composed of hierarchical classification of shots composing the scene, and Structure whose type is FromBrowsingNode, in which the node indicates a cluster of similar shots for example and the link indicates an AVGraph which depicts a transition between groups. Note that SCN in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 32, represents a descriptor code. Also, the Abstract enables to give the

following abstract to the scene. Namely, the abstract is similarity including shots or AVPrototype as its elements.

[0130]

The scene descriptor denotes a sequence of shots semantically related.

[0131]

With use of the scene descriptor, the content of a scene of video data is specifically described, and the scene is intended to be described with inheriting Browsing Node descriptor. More specifically, with use of the shot descriptor, a shot as data, based on which the clustering for detecting a scene, as will be further described later.

[0132]

Further, the description format for the topic descriptor is as shown in FIG. 33. The topic descriptor includes the browsing node descriptor as the Superclass. Also, the topic descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Abstract whose type is FromAVSegment, Synopsis whose type is FromBrowsingNode and which indicates the possible outline to a topic composed of synopsis-type data structured by a sequence of scenes, Index whose type is FromBrowsingNode and which indicates an index to the topic composed of hierarchical classification of scenes structuring the topic, and Structure whose type is FromBrowsingNode, in which the node indicates a cluster of similar scenes for example and the link indicates an AVGraph which depicts a transition between groups.

Note that TPC in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 33, represents a descriptor code. Also, the abstract can carry out the following abstraction. Namely, the first abstraction is (temporal) similarity including scenes or AVPrototypes as its elements. The second abstraction is (temporal) similarity including shots as value. The third abstraction is (temporal) similarity including audio frames and visual frames as value.

[0133]

The topic descriptor denotes a sequence of shots semantically related.

[0134]

With use of the topic descriptor, the content of a topic of video data is described specifically, and the scene is intended to be described with inheriting Browsing Node descriptor.

[0135]

Further more, the description format for the program descriptor is as shown in FIG. 34. The program descriptor includes the browsing node descriptor as the Superclass. Also, the program descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Abstract whose type is FromAVSegment, Synopsis whose type is FromBrowsingNode and which indicates that abstract composed of a sequence of scenes, abstract composed of a sequence of shots and abstract structured by highlight scenes of a program are included as abstract of the program, Index whose type is FromBrowsingNode, and Structure whose type

is FromBrowsingNode, in which the node indicates a cluster of similar topics for example and the link indicates an AVGraph which depicts a transition between groups. Note that PRG in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 34, represents a descriptor code. Also, the abstract can carry out the following abstraction. Namely, the first abstraction is (temporal) similarity including topics as its elements. The second abstraction is (temporal) similarity including scenes as value. The third abstraction is (temporal) similarity including shots as value. Further, the Index to the program is formed from an index for various levels such as hierarchical index to all shots and/or hierarchical index to all scenes and/or hierarchical index to all topics.

[0136]

The program descriptor denotes a program composed of a sequence of topics.

[0137]

In addition, the description format for the highlight descriptor is as shown in FIG. 35. The highlight descriptor includes the browsing node descriptor as the Superclass. Also, the highlight descriptor has, as features denoting a descriptor or a sort of numerical value of lower-level element, Interest whose type is Weight, which indicates estimated interest level to the highlight, and which indicates that larger value indicates a higher interest level. Note that HI in a column of a descriptor which denotes the name of the descriptor and its descriptor code shown in FIG. 35, represents a descriptor code.

[0138]

This highlight descriptor grasps an occurrence assumed to be some highlight occurred in a broadcasting program. For example, this highlight descriptor is used for marking a home run in a baseball game, a particularly impressive moment in a movie or the like. An attribute which indicates how much the highlight is interesting or impressive for the audience, is attached to each highlight.

[0139]

In the description scheme composed of a set of aforementioned descriptors, by using information included in the indices in various browsing nodes, it becomes possible to structure at various detailed levels audio-visual indices of video data having a hierarchically structure. For example, in a program, for all shots included in video data, a hierarchical structure classed based on its similarity is structured. When it is desired to display information on a certain detailed level based on the above, the hierarchical structure should be divided at the designated level and only the upper classification should be used. As a result, the classification has a lower-classified hierarchical structure. Selecting each representative member of the groups from these lower-classified groups respectively, respective abstract of visual and audio of the group members can be displayed.

[0140]

Further, in description scheme, use of a synopsis related to each level of a hierarchical video structure makes it possible to pick out a hierarchical audio-visual

table of the video data. For example, when there is a synopsis as shown below in Table 1 at each level and it is desired to display the table, repeated extraction of key frames at each level from representative key frames of each shot makes it possible to pick out an audio-visual table.

[0141]

[Table 1]

Synopsis for levels of video structure

Level	Synopsis
Shot	Sequence of key frames
Scene	Sequence of key shots
Topic	Sequence of key scenes
Program	Sequence of key scenes

[0142]

Concrete processing for extraction of a structure from video data, and, in correspondence with such processing, feature data which describes a extracted video structure or the like generated by using a describing procedure defined as the aforementioned description scheme, will be described below. Note that the process for extraction of a structure from video data can be effected by both the transmitter 10 and receiver 20 in the data transmitting-receiving processing system having previously been described. Extraction of a scene structure from video data in the transmitter 10,

generation of feature data corresponding to the extraction, and transmission of the feature data with the video data to the receiver 20, will be described hereinbelow.

[0143]

Using the features of the visual segments and audio segments in the video data, the transmitter 10 measures a similarity between the segments and groups these segments into scenes, thereby to extract a video structure automatically. At this time, since repeated similar segments lying close to each other on the time base are parts of the generally same scene, the transmitter 10 detects the scenes by detecting and grouping such segments. The transmitter 10 carries out a series of processes as outlined in FIG. 36, thereby to detect scenes.

[0144]

First at step S1, the transmitter 10 carries out video segmentation as shown in FIG. 36. More particularly, the transmitter 10 divides an input video data into either visual segments or audio segments, or possibly into both segments. The transmitter 10 sets no specific precondition to the video segmenting method applied thereto. For example, the transmitter 10 segments video data by the method disclosed in "G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, J. of Visual Communication and Image Representation 7: 28-4, 1996". The video segmenting method of this sort is well known in this field of art. Any video segmenting method can be applied to the transmitter 10.

[0145]

Next at step S2, the transmitter 10 extracts a feature of a segment. More specifically, the transmitter 10 calculates a set of features for each segment, and represents the feature of the segment. The transmitter 10, for example, calculates a time duration of each segment, features concerning video or visual such as color histogram and texture feature, frequency analysis result, features concerning audio such as level and pitch, activity determination result, and the like, as applicable features. Of course, the transmitter 10 is not limited to these applicable features.

[0146]

Next at step S3, the transmitter 10 carries out similarity measurement between segments using their features. More specifically, the transmitter 10 carries out dissimilarity measurement between segments and determines how similar the two segments are to each other according to the criterion for measurement. Using the features having been extracted at step S2, the transmitter 10 calculates a criterion for measurement of dissimilarity.

[0147]

At step S4, the transmitter 10 groups the segments. More particularly, using the dissimilarity measurement criteria calculated at step S3 and features extracted at step S2, the transmitter 10 iteratively groups similar segments lying close to each other on the time base, and superposes these segments to merge to the group.

Thus, the transmitter 10 outputs a finally generated group as a scene.

[0148]

With the above series of processes, the transmitter 10 can detect a scene from video data. Therefore, using the above result, the user can summarize the content of the video data and quickly access to interesting points in the video data.

[0149]

The processing in the transmitter 10 at each of the steps shown in FIG. 36 will further be described in detail below.

[0150]

First the video segmentation at step S1 will be discussed below. The transmitter 10 divides input video data into either visual or audio segments, or into both segments if possible. Many techniques are available for automatic detection of a boundary between segments in the video data. As mentioned above, no specific precondition is set to the video segmenting method in the transmitter 10. However, in the transmitter 10, the accuracy of scene detection in the later step essentially depends on the accuracy of the video segmentation being a basis. Note that in the transmitter 10, the scene detection can tolerate some error in the video segmentation. Particularly, in this transmitter 10, it is preferable that an error occurs more in excessive segment detection than insufficient one for the video segmentation. As long as similar segments are detected excessively, or even the segment detection is not excessive, segments being detected excessively can generally be grouped as those included in the same scene at the scene detection.

[0151]

Next the feature detection at step S2 will be discussed below. The features are attributes of segments, characterizing the contents of the segments and providing data according to which a similarity between different segments is measured. The transmitter 10 calculates a set of features for each segment and represents its characteristic. The transmitter 10 does not depend on any specific detail of the features. The features considered to be effectively usable in the transmitter 10 include visual feature, audio feature and visual- audio feature as will be described below. The requirement for these features usable in the transmitter 10 is that they must include dissimilarity measurement. For a higher efficiency, the transmitter 10 simultaneously effects, in practice, a feature extraction and the aforementioned video segmentation as the case may be. The features which will be described below enable to carry out the above processing.

[0152]

As for features, one concerning video is given first. A video (image) which forms a segment, represents most of describing content of the segment. Therefore, the similarity of the visual segment often can be changed to similarity of the video itself. Thus, the feature concerning visual is one of the important features usable in the transmitter 10. Since the feature concerning the visual denotes static information rather than dynamic information, the transmitter 10 extracts the feature concerning visuals in the visual segment so as to obtain dynamic information by a method which will be described later.

[0153]

Although there are many well-known features concerning visual, the transmitter 10 uses a color feature (histogram) and visual correlation as features concerning a video.

[0154]

In the transmitter 10, colors of a video are important materials for determining whether or not two videos are similar to each other. Determining a similarity between videos with use of a color histogram is well known as disclosed in, for example, "G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, J. of Visual Communication and Image Representation 7: 28-4, 1996". The color histogram is acquired by dividing a three-dimensional ColorSpace such as HSV, RGB or the like for example into n areas and calculating a relative ratio of pixels of a video in each area. Also, information thus acquired gives an n-dimensional vector. In regard to a compressed video data, a color histogram is extracted directly from compressed data as disclosed in the U. S. Patent No. 5,708,767.

[0155]

The transmitter 10 samples an original YUV ColorSpace in images forming a segment at a rate of 2 bits per color channel, thereby obtains a histogram with length of $2^{2\cdot3} = 64$.

[0156]

Such a histogram represents a total color tone of an video but includes no timing

data. For this reason, a video correlation is calculated as another visual feature in the transmitter 10. For the scene detection in the transmitter 10, the interleaved segment gives an influential index of scene structure. The transmitter 10 sub-samples a video to grayscale images each of $M \times N$ in size, so as to calculate a video correlation. Note that both M and N are small values, for example, $M \times N$ may be 8×8. A small grayscale image is interpreted as a feature of MN in size.

[0157]

The features different from the above-mentioned feature concerning video is a one concerning sound. The feature concerning audio can be used for representing the content of an audio segment. The transmitter 10 uses a frequency analysis, pitch, level as features concerning audio. These audio features are known from various documents.

[0158]

First, the transmitter 10 can determine the distribution of frequency information in a single audio frame, by making a frequency analysis of a Fourier Transform or the like. For example, the transmitter 10 can use FFT (Fast Fourier Transform;) component, frequency histogram, power spectrum and other features to represent a distribution of frequency information over a single audio frame.

[0159]

Also, the transmitter 10 may use pitches such as a mean pitch and maximum pitch, and levels such as mean loudness and maximum loudness, as effective audio for

representation of audio segments.

[0160]

Further features are those common to visual and audio segments. They are neither any visual feature nor audio feature, but provide useful information for representation of contents of segments in a scene. The transmitter 10 uses a segment length and an activity, as features concerning common visual-audio.

[0161]

The transmitter 10 can use a segment length as a feature concerning common visual-audio. The segment length is a time length in a segment. A scene has a rhythm feature of the change, and involves the similar tendency to the one of the segment length in the scene. For example, short segments contiguous to each other with a short time between them represent a commercial program. On the other hand, segments included in a conversation or dialogue scene are even longer, but similar to each other in the length. The transmitter 10 can use segment length including such characteristics, as a feature concerning common visual-audio.

[0162]

Also, the transmitter 10 uses an activity as a feature concerning common visual-audio. The activity indicates how dynamic or static the content of a segment are felt. For example, if the activity is visually dynamic, it indicates a degree of rapidity with which a camera moves along an object or a degree of rapidity with which an object being shot.

[0163]

The activity is indirectly calculated by measuring a mean inter-frame dissimilarity in a feature such as color histogram. If dissimilarity measurement criterion for the feature F measured between the frames i and j, are defined as d_F (I, j), a video activity V_F is given by the following equation (1).

[0164]

[Equation 1]

$$V_F = \frac{\sum_{i=b}^{f-1} d_F(i, i+1)}{f - b + 1} \quad \cdots\cdots \quad (1)$$

[0165]

In the equation (1), b and f are respective numbers for a first frame and last frame in one segment. More specifically, the transmitter 10 calculates the video activity V_F by using the above-mentioned histogram for example.

[0166]

Meanwhile, the features including the above-mentioned features concerning audio basically indicate static information of a segment as in the above. To accurately represent the feature of a segment, however, dynamic information has to be represented. For this reason, the transmitter 10 represents dynamic information by a feature sampling method which will be described below.

[0167]

As shown in FIG. 37 for example, the transmitter 10 extracts more than one set of static feature value, starting at different time points in one segment. At this time, the transmitter 10 determines the number of features to extract by keeping a balance between a maximization of fidelity and a minimization of redundancy. For example, when a certain image in the segment is designated as a key frame, a histogram calculated from the key frame will be a feature to extract.

[0168]

Using a sampling method, the transmitter 10 determines which of the samples are to be selected from all value of a feature in a segment. The transmitter 10 needs an excellent sampling method.

[0169]

Here, it will be considered that a certain sample is always selected at a predetermined time point, for example, at the last time point in a segment. In this case, samples from two arbitrary segments fading into black will be same black frames, so that the same feature value will possibly be acquired. That is, two sampled frames will be determined to be extremely similar to each other, whatever these video image contents are. Such problem will take place since the samples are not good representative values.

[0170]

For this reason, the transmitter 10 extracts a statistically representative value,

not does a feature at such a fixed point. Here, the general feature sampling method will be described concerning two cases that a feature can be represented as a real-number n-dimensional vector, as being the first case, and only a dissimilarity measurement criterion can be used, as being the second case. It should be noted that the first case includes best-known features concerning visual and audio such as histogram, power spectrum, etc.

[0171]

In the first case, the number of samples is predetermined to be k and the transmitter 10 automatically segments a feature value of an entire segment into k different groups by using the well-known k-means clustering method as disclosed in "L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John-Wiley and Sons, 1990". And then, the transmitter 10 selects one sample from each of the k groups. That is, the transmitter 10 selects a centroid (mean vector) from each of the groups or a sample approximate to the centroid from each of the groups. The transmitter 10 can carry out this processing in a short time, so that it requires just the linear time for the number of samples.

[0172]

On the other hand, in the second case, the transmitter 10 forms the k groups by the use of the k-medoids algorithm method also disclosed in "L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John-Wiley and Sons, 1990". The transmitter 10 uses, as a sample value, a medoid of the group,

which is similar to the above-mentioned centroid of the groups, for each of the groups.

[0173]

Thus, the transmitter 10 can extract static features to represent dynamic information.

[0174]

As in the above, the transmitter 10 can extract various features. Each of such features is generally insufficient for representing, by itself, a content of a segment. For this reason, the transmitter 10 can select a set of mutually complementary features by combining these various features. For example, the transmitter 10 can obtain more information than that of each feature by combining the above-mentioned color histogram and video image correlation with each other.

[0175]

Next, the similarity measurement of between segments, in which the features at step S3 in FIG. 36 are used, will be described below. Using the dissimilarity measurement criterion being a function to calculate a real-number value with which it is determined how dissimilar two features are to each other, the transmitter 10 carries out a similarity measures between the segments. When the dissimilarity measurement criterion is small, it indicates that two features are similar to each other. If the criterion is large, it indicates that the two features are not similar to each other. The function for calculation of the dissimilarity between the two segments S_1 and S_2 concerning the feature F is defined as dissimilarity measurement criterion $d_F(S_1, S_2)$.

This function has to meet the characteristics shown in the equations (2) below.

[0176]

[Equation 2]

$$d_F(S_1, S_2) = 0 \qquad \text{(when } S_1 = S_2\text{)}$$

$$d_F(S_1, S_2) \geq 0 \qquad \text{(as for all the } S_1, S_2\text{)}$$

$$d_F(S_1, S_2) = d_F(S_2, S_1) \qquad \text{(As for all the } S_1, S_2\text{)} \qquad ....(2)$$

[0177]

It should be noted that some appropriate dissimilarity measurement criterion is only applicable to a specific feature. However, as disclosed in "G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, J. of Visual Communication and Image Representation 7: 28-4, 1996", and "L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John-Wiley and sons, 1990", many general dissimilarity measurement criteria are known as being applicable to the measurement of a similarity between features represented as points in a n-dimensional space. Such criteria are a Euclidean distance, inner product, L1 distance, etc. Particularly, detected that the L1 distance act effectively on various features including the histogram, image correlation, etc., the transmitter 10 finds the L1 distance between two of n-dimensional vectors A and B by the following equation (3).

[0178]

[Equation 3]

$$d_{L1} = (A,B) = \sum_{i=1}^{n} |A_i - B_i| \qquad .... (3)$$

[0179]

There, the subscript indicates the i-th element of the n-dimensional vector.

[0180]

As mentioned above, the transmitter 10 extracts feature values at various points in the segment, regarding features which change depending on the time. Then, to determine a similarity between two extracted features, the transmitter 10 defines dissimilarity of the extracted feature on the basis of a dissimilarity measurement criterion for the feature on which dissimilarity is based. In many cases, it is most advantageous that the dissimilarity should be determined, using a pair of dissimilarity values of the feature including the least dissimilarity, the feature selected from two extracted features respectively. Namely, the transmitter 10 determines the minimum dissimilarity. In this case, the sampled dissimilarity measurement criterion between two extracted features SF_1 and SF_2 is defined as the following equation (4).

[0181]

[Equation 4]

$$d_S(SF_1, SF_2) = \min_{F_1 \in SF_1, F_2 \in SF_2} d_F(F_1, F_2) \qquad .... (4)$$

[0182]

The function d_F (F_1, F_2) in the equation (4) above indicates a dissimilarity measurement criterion for the extracted features F on which the equation (4) is based. Also, it is more preferable that the maximum or mean value of the dissimilarity is taken instead of a maximum value, as the case may be.

[0183]

Further, in many cases, it is necessary for the transmitter 10 to combine information derived from many features for the same segment. The transmitter 10 calculates a weighted combination of various feature vector dissimilarity functions, as one of the method. That is, when there are available k features F_1, F_2, ..., F_k, the transmitter 10 uses a dissimilarity measurement criterion d_F (S_1, S_2) for combined features which are given by the following equation (5).

[0184]

[Equation 5]

$$d_F(S_1,S_2) = \sum_{i=1}^{k} w_i d_{Fi}(S_1,S_2) \qquad \text{.... (5)}$$

[0185]

Here, $\{w_i\}$ is a pair of weighting and $\Sigma_i w_i = 1$.

[0186]

As in the above, the transmitter 10 can calculate a dissimilarity measurement criterion using features having been extracted at step S2 in FIG. 33 to determine a

similarity of segments.

[0187]

Next, the segment grouping at step S4 in FIG. 36 will be described below. Using the dissimilarity measurement criterion and extracted features, the transmitter 10 repeatedly groups similar segments lying close to each other on the time basis, and outputs a finally produced group as a scene.

[0188]

When detecting a scene by grouping segments, the transmitter 10 carries out two of basic processing. As the first processing, the transmitter 10 detects groups of similar segments lying close to each other on the time base. Most of the groups acquired by the processing, will be a part of the same scene. Since the segments of the transmitter 10 are overlapped in time, it gathers together similar scenes in which the time is overlapped, as the second processing. The transmitter 10 starts these processing with independent segments, and repeats them. Then the transmitter 10 organizes a step-by-step larger group of segments and outputs a finally produced group as a set of scenes.

[0189]

To control these processing, the transmitter 10 is controlled under the following two constraints.

[0190]

Namely, under one of the two constraints, the transmitter 10 has to adopt a

dissimilarity threshold δ_{sim} to determine how much mutually similar two segments should be, so as to be considered that the two segments are sufficiently similar to regarded as the same scene. As shown in FIG. 36 for example, the transmitter 10 judges whether one of the segments belongs to similarity range or dissimilarity range to the other.

[0191]

It should be noted that the transmitter 10 may be adapted to set the dissimilarity threshold δ_{sim} by the user or automatically.

[0192]

Under the second constraint, the transmitter 10 has to adopt a temporal threshold T to determine how much tow segments should be divided on the time base, based on which the two segments can be considered to be included in the same scene. As shown in FIG. 39 for example, the transmitter 10 puts, into the same scene, two similar segments A and B lying close to each other within the temporal threshold T but not two segments B and C having between them a time gap not within the temporal threshold T. Thus, because of the constraint by the temporal threshold T value, the transmitter 10 will not erroneously put two segments similar to each other but largely apart from each other, into the same scene.

[0193]

It is assumed herein that the transmitter 10 adopts the hierarchical clustering method disclosed in "L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An

Introduction to Cluster Analysis, John-Wiley and Sons, 1990", so as to acquire a group of similar segments. In this algorithm, a criterion d_C (C_1, C_2) for determination of a dissimilarity between two Clusters C_1 and C_2 is defined as a minimum dissimilarity between two included elements, as shown in the following equation (6).

[0194]

[Equation 6]

$$d_C(C_1, C_2) \cong \min_{S_1 \in C_1, S_2 \in C_2} dist_s(S_1, S_2) \qquad \text{.... (6)}$$

[0195]

It should be noted that in the transmitter 10, a minimum function expressed by the equation (6) can easily be replaced with a maximum function or mean function.

[0196]

First at step S11 in FIG. 40, the transmitter 10 initializes a variable N to the number of segments. The variable N always indicates the concurrent number of groups detected.

[0197]

Next at step S12, the transmitter 10 generates a set of Clusters. In the initial state, the transmitter 10 regards each of N segments as different Clusters. That is, there exist N Clusters in the initial state. Each of the Clusters has features indicating the start time and end time represented by C^{start} and C^{end}, respectively. A set of

elements included in each Cluster is arranged in order based on the start time C^{start}.

[0198]

Further at step S13, the transmitter 10 initializes a variable t to 1. At step S14, the transmitter 10 judges whether or not the variable t is larger than the temporal threshold T. Here, if the variable t is larger than the temporal threshold T, the transmitter 10 will shift its processing to step S23. When the variable t is smaller than the temporal threshold T, it will shift its processing to step S15. Since the variable t is 1, however, the transmitter 10 will shift its processing to step S15.

[0199]

At step S15, the transmitter 10 calculates the dissimilarity measurement criterion d_C to detect two of the N Clusters that are the most similar to each other. Since the variable t is 1, however, the transmitter 10 will calculate the dissimilarity measurement criterion d_C between adjacent Clusters to detect among the adjacent Clusters a pair of Clusters that are the most similar to each other.

[0200]

A method for detecting two Clusters which are the most similar to each other can be considered to scan all possible pairs of object Clusters. The transmitter 10 includes time constraint by the temporal threshold T, so that it can limit the number of a set of the object Clusters. Therefore, the transmitter 10 should scan only Clusters segmented by the t segments. Since sets of Clusters are arranged in the temporal order, the transmitter 10 scans some segment in its forward or rear direction. Once

transmitter 10 scans a segment which is apart from the t segments, the scan processing can be completed such that all segments after the scanned segment are out of object.

[0201]

The two Clusters thus detected are defined as C_i and C_j, respectively, and dissimilarity value between the Clusters C_i and C_j is defined as d_{ij}.

[0202]

At step S16, the transmitter 10 judges whether the dissimilarity d_{ij} is larger than the dissimilarity threshold δ_{sim}. Here, if the dissimilarity d_{ij} is larger than the dissimilarity threshold δ_{sim}, the transmitter 10 will shift its processing to step S21. When the dissimilarity dij is smaller than dissimilarity threshold δ_{sim}, it will shift its processing to step S17. It is assumed here that the dissimilarity d_{ij} is smaller than the dissimilarity threshold δ_{sim}.

[0203]

At step S17, the transmitter 10 merges the Cluster C_j into the Cluster C_i. That is, the transmitter 10 adds to the Cluster C_i all the elements in the Cluster C_j.

[0204]

Next at step S18, the transmitter 10 removes the Cluster C_j from the set of Clusters. It should be noted that if the start time C_i^{start} changes due to the combination of the two Clusters C_i and C_j, the transmitter 10 will rearrange the elements in the set of Clusters so as to maintain the order set at the start time.

[0205]

Further at step S19, the transmitter 10 subtracts 1 from the variable N.

[0206]

At step S20, the transmitter 10 judges whether the variable N is 1 or not. Here, if the variable N is judged to be 1, the transmitter 10 will shift its processing to step S23. When the transmitter 10 determines that the variable N is not 1, it will shift its processing to step S15. It is assumed here that the variable N is not 1.

[0207]

Thus, at step S15, the transmitter 10 calculates the dissimilarity measurement criterion d_C again to detect two Clusters being the most similar to each other from the N - 1 Clusters. Since the variable t is also 1 here, the transmitter 10 calculates the criterion d_C for determination of the dissimilarity between adjacent Clusters to detect a pair of Clusters being the most similar to each other from the Clusters.

[0208]

Next at step S16, the transmitter 10 judges whether the dissimilarity d_{ij} is larger than the dissimilarity threshold δ_{sim}. It is also assumed here that the dissimilarity d_{ij} is smaller than the dissimilarity threshold δ_{sim}.

[0209]

The transmitter 10 carries out the processing at steps S17 to S20.

[0210]

When as a result of the repetition of the above processing and subtraction of 1 from the variable N, it is determined at step S20 that the variable N is 1, the transmitter

10 will merges Clusters each including a single segment at step S23. That is, the transmitter 10 does not need to carry out the above processing but terminates the series of processing by grouping all segments into one Cluster.

[0211]

If the transmitter 10 determines at step S16 that the dissimilarity d_{ij} is larger than the dissimilarity threshold δ_{sim}, it will shift its processing to step S21. In this case, it will repeatedly merge Clusters which are overlapped on the time basis at step S21. Namely, if the time interval between [C_i^{start} and C_i^{end}] of the Cluster C_i is concurrent with that between [C_j^{start} and C_j^{end}] of the Cluster C_j, the two Clusters C_i and C_j overlap each other on the time base. Thus, the transmitter 10 can arrange the Clusters in a set based on the start time of the Cluster set to detect overlapping Clusters and combine the Clusters together.

[0212]

At step S22, the transmitter 10 adds 1 to the variable t which will thus be t = 2, and shifts its processing to step S14 where it judges whether or not the variable t is larger than the temporal threshold T. It is also assumed here that the variable t is smaller than the temporal threshold T and the transmitter 10 shifts its processing to step S15.

[0213]

At step S15, the transmitter 10 calculates the dissimilarity measurement criterion d_C and detects two of a plurality of Clusters existing currently, that two are

the most similar to each other. However, since the variable t is 2, the transmitter 10 calculates the criterion d_C for determination of the dissimilarity between Clusters being every other Cluster to detect a pair of Clusters the most similar to each other from the Clusters.

[0214]

Then at step S16, the transmitter 10 judges whether the dissimilarity d_{ij} between Clusters C_i and C_j existing every other Cluster, is larger than the dissimilarity threshold δ_{sim}. It is assumed here that the dissimilarity d_{ij} is smaller than the dissimilarity threshold δ_{sim}. Via processing at steps S21 and S22, the transmitter 10 adds 1 to the variable t which will thus bet = 3, and moves to processing at step S14 and subsequent steps. When the variable t is 3, the transmitter 10 obviously calculates, at step S15, the criterion d_C for determination of the dissimilarity between Clusters existing every two Clusters, and detects a pair of Clusters which are the most similar to each other from the Clusters.

[0215]

When as a result of the repetition of the above processing and addition of 1 to the variable t, the transmitter 10 determines at step S14 that the variable t is larger than the tine threshold T, it shifts its processing to step S23 and combines Clusters each including a single segment there. That is, the transmitter 10 regards discrete Clusters as ones each including only single element. If there exist a sequence of such Clusters, the transmitter 10 will merge them together. This process combines together any

simple scenes with segments having no relation in similarity. Noted that the transmitter 10 has not to always carry out this process.

[0216]

The transmitter 10 terminates this series of processing via the processing at step S22.

[0217]

With this series of processing, the transmitter 10 can gather the plurality of Clusters and generate a scene.

[0218]

When a scene is detected, the transmitter 10 can also use a weighting function to combine a variety of dissimilarity measurement criteria for different types of features, as well as the use of a single dissimilarity measurement criterion, so as to judge whether segments in pairs belong to the same group, as having previously been described. In the transmitter 10, such weighting for the features can only be obtained after trial and error, and when the features are different in type from each other, it is usually difficult to appropriately weight them. However, for combining a color histogram and texture feature, for example, the transmitter 10 can detect possible scenes for these features and combine each of detected scene structure with a single scene structure. Each of the results of scene detection for the features will be referred to as "scene layer" hereinafter. For example, when a color histogram and segment length are used as features, the transmitter 10 can detect scenes to the features, to

obtain a scene layer for the color histogram and a one for the segment length, and combine these scene layers into a single scene structure.

[0219]

Further, information from video and audio domains cannot be combined in principle. However, the transmitter 10 can combine scene layers obtainable from video and audio domains into a single scene structure, by using a similar method to that for combining structures having different types in quality from each other.

[0220]

Such processing will be described below. It is assumed here that there are k features F_1, F_2, ..., F_k each representing one similarity criterion and there are available a dissimilarity measurement criterion d_F^i, dissimilarity threshold δ^i_{sim} and a temporal threshold T^i correspondingly to each of the features F_i. Using the dissimilarity measurement criterion d_F^i, dissimilarity threshold δ^i_{sim} and a temporal threshold T^i for each of these features F_i, the transmitter 10 detects a set of scene layers $X_i = \{X_i^j\}$. For example, the transmitter 10 is assumed to detect divisional scene layers for video and audio information, respectively, and generate two independent scene layers $X_i = \{X_i^j\}$ ($i = 1, 2$) for the video and audio information, respectively.

[0221]

The transmitter 10 has to determine how to combine scene boundaries for combination of difference scene layers into a single scene structure. It is not guaranteed that the scene boundaries do not always match one another. It is assumed

here that for each of the scene layers, there are boundary points $t_{i1}, t_{i2}, \ldots, t_i | X_i |$ represented by a sequence of times indicating the scene boundaries. The transmitter 10 first selects a certain scene layer to be a basis for alignment in order to combine various scene layers into a single group. Then, the transmitter 10 determines whether or not the scene layer is a scene boundary in the scene structure generated by finally combining, for each of the boundary points $t_{i1}, t_{i2}, \ldots, t_i | X_i |$.

[0222]

Here, for each i being a number for a set of scene layers, $B_i(t)$ is assumed to be a Boolean function indicating whether a scene boundary "near" a certain time t exists or not in the i-th scene layer X_i. The meaning of term "near" varies, and it can be for example 0.5 sec, in case of visual and audio information being prepared.

[0223]

The transmitter 10 calculates the result of the function $B_1(t_j)$ for each of the boundary points $t_j = t_{ij}$, $j = 1, \ldots, | X_j |$ and $i = 1, \ldots, k$. The calculation result indicates whether the scene boundary exists near the time t in the scene X_i for each of independent scene layers. The transmitter 10 uses $B_i(t_j)$ as a decision function, so as to determine whether or not the time t is a scene boundary in the combined determination.

[0224]

A simple example of the decision function of this sort, is to calculate the sum of the value such that $B_i(t_j)$ is equal to 1 and the constant m and over, and the time

point t expresses the scene boundary of the final scene structure. Particularly, when m = 1, it means same as "or-ing" boundary point. When m = k, it means same as the requirement for preparing all boundary points.

[0225]

Thus, the transmitter 10 can combine difference scene layers into a single scene structure.

[0226]

The processing in the transmitter 10 can be described using the aforementioned descriptors.

[0227]

The descriptors used to describe the feature extraction process at step S2 in FIG. 36 includes for example the ActivityLevel descriptor, Signature descriptor, TemporalSignature descriptor, Synopsis descriptor, etc.

[0228]

The ActivityLevel descriptor is used to describe activities as visual and audio features at the feature extraction process. Also, the Signature descriptor is used to describe a feature sampling method for describing dynamic information in the process of feature extraction. Further, the TemporalSignature descriptor can be used at the process to search the best feature for representing dynamic information at the feature extraction process. Moreover, the Synopsis descriptor is used for that the scene detected at step S4 in FIG. 36 is to be represented by representative frames detected

by the extraction of dynamic feature in the process of feature extraction.

[0229]

Also, the descriptors used for description of the process of segment similarity measurement, using the features at step S3 in FIG. 36 include the WeightVector descriptor, etc.

[0230]

The WeightVector descriptor is used to weight for combining information from a plurality of features in the process of feature extraction.

[0231]

Further, the descriptors used to describe the process of segment grouping at step S4 in FIG. 36 include for example the AVPrototype descriptor, Cluster descriptor, CompositeSegment descriptor, AVGraph descriptor, Shot descriptor, Scene descriptor, etc.

[0232]

The AVPrototype descriptor and Cluster descriptor are used to describe the clustering processing in the process of grouping. Also, the CompositeSegment descriptor is used to describe the result of clustering segments and scenes at the step of grouping. Further, the AVGraph descriptor is used to describe the content of scenes detected at the step of grouping as a repeated structure of the scenes. Further more, the Shot descriptor is used to specify the shots, on which the scene detection at the grouping step is based, and Scene descriptor is used to specify the scenes detected at

the grouping step.

[0233]

Therefore, a description scheme for describing a scene structure in a certain video data is represented by a set of plural descriptors. More specifically, a description scheme for describing a certain scene and feature data are shown in FIGS. 41 and 42. It is assumed here that a scene to be described starts at a frame No. "12547" and ends at a frame No. "12830". Note that in FIG. 42, the description following a semi-colon (;) indicates the reference descriptor in the line.

[0234]

First, the feature data indicates the beginning of a descriptor specifying a scene description in the line 1 in FIG. 42. The node on the description scheme corresponding to this scene is the apex node in FIG. 41.

[0235]

Next in the lines 2 and 3 in FIG. 42, the feature data respectively specifies startPoint and endPoint of a scene with a time, frame number or the like. The information on the description scheme is represented by startTime and endTime in FIG. 41.

[0236]

Also in the line 4 in FIG. 42, the feature data specifies a location where the scene exists in the video contents.

[0237]

Further in the line 5 in FIG. 42, the feature data specifies the beginning of the Synopsis descriptor, and designates a KEY_FRAME for a user's browsing.

[0238]

Next, in the line 6 in FIG. 42, the feature data specifies the beginning of the Sequence descriptor, and that there is a sequence of AVElements subsequently. The AVElement is herein a key frame.

[0239]

Also, in the lines 7, 8 and 9 in FIG. 42, the feature data respectively designates the key frames for the first, second and third synopses as KEY_FRAME 1, KEY_FRAME 2 and KEY_FRAME 3, respectively. Although KEY_FRAME 1, KEY_FRAME 2 and KEY_FRAME 3 are not concretely specified, they need to be designated anywhere in practice.

[0240]

Moreover, in the lines 10 and 11 in FIG. 42, the feature data specifies respectively the end of the Sequence descriptor and that of the Synopsis descriptor.

[0241]

Next, in the line 12 in FIG. 42, the feature data specifies the beginning of the Abstract descriptor to designate a representative part of the video content.

[0242]

Further, in the line 13 in FIG. 42, the feature data specifies the beginning of the Signature descriptor to designate a part of the scenes used for clustering and the like.

It is herein declared that the beginning is specified with SHOT_SIGNATURE.

[0243]

Moreover, the feature data specifies start of designating the members of Signature in the line 14 in FIG. 42.

[0244]

The feature data specifies the first, second and third members as shot1, shot2 and shot3, respectively. The shot1 and shot2 are defined in the lines 22 and 32.

[0245]

Also, the feature data specifies end of designating the Signature members in the line 17 in FIG. 42.

[0246]

Next, the feature data designates weighting of the Signature members in the line 18 in FIG. 42. It is specified that the shot1 has importance of 0.1 while the shot2 and shot3 have importance of 0.45, respectively.

[0247]

Then, the feature data specifies ends of the Signature descriptor and Abstract descriptor respectively in the lines 19 and 20 in FIG. 42.

[0248]

Next, the feature data specifies beginning of the Components descriptor in the line 21 in FIG. 42 to designate components of the scene. The nodes on the description scheme corresponding to the components are the ones a step lower than the node

which denotes the scene.

[0249]

In the line 22 in FIG. 42, the feature data specifies beginning of the Shot descriptor to designate the first component of the scene as the shot1, and declares that the content will be described subsequently.

[0250]

Further in the lines 23 and 24 in FIG. 42, the feature data specifies start and end points of the shot1 respectively by a time, frame number or the like. The startPoint is the frame No. "12547" and the endPoint is the frame No. "12616" in this example. This information on the description scheme is specified represented by "startTime" and "endTime" in a node being one step lower than the node specifying the component in FIG. 41.

[0251]

Next, the feature data specifies beginning of the Abstract descriptor in the line 25 in FIG. 42 to designate a representative part of the content of the shot1.

[0252]

Also, in the line 26 in FIG. 42, the feature data specifies beginning of the Signature descriptor to designate a part of the shot1 for use of clustering and the like. It is declared here that the Signature is specified by FRAME_SIGNATURE.

[0253]

Further in the line 27 in FIG. 42, the feature data specifies that the Signature has

only one member, to designate the frame number. It is described that the frame number is "12590".

[0254]

In the line 28 in FIG. 42, the feature data designates weighting the members of the Signature as descriptor. Since the Signature has only one member, the weight will be 1.0.

[0255]

Then, the feature data specifies end of the Signature descriptor, end of the Abstract descriptor and end of the Shot descriptor in the lines 29, 30 and 31, respectively, in FIG. 42.

[0256]

Next in the line 32 in FIG. 42, the feature data specifies beginning of the Shot descriptor to designate that the second component of the scene is "shot2", and declares that the content of the shot will be described subsequently.

[0257]

In the lines 33 and 34 in FIG. 42, the feature data specifies start and end points of the Shot2 by a time, frame number or the like. The start point is specified here to be the frame No. "12617" while the end point is to be the frame No. "12628". This information on the description scheme is specified by "startTime" and "endTime" in a node being one step lower than the node specifying the component in FIG. 41.

[0258]

The feature data describes many descriptors specifying the content of the shot2 in the line 35 in FIG. 42, but they are omitted herein.

[0259]

In the line 36 in FIG. 42, the feature data specifies end of the Shot descriptor.

[0260]

In the lines 37 and 38 in FIG. 42, the feature data specifies various descriptors specifying the contents of various shots and many descriptors specifying each of the shots respectively, but they are omitted herein.

[0261]

In the line 39 in FIG. 42, the feature data specifies end of the Components descriptor.

[0262]

Next in the line 40 in FIG. 42, the feature data declares that specifies some AVHierarchy descriptors specifying the index of the scenes will be described subsequently.

[0263]

In the line 41 in FIG. 42, the feature data specifies the beginning of the Hierarchy descriptor to declare that the hierarchy is specified by SHOT_INDEX.

[0264]

Further, in the line 42 in FIG. 42, the feature data specifies the beginning of the Cluster descriptor to declare that the first Cluster is described subsequently and the

name of this Cluster is "Cluster1".

[0265]

Next, in the line 43 in FIG. 42, the feature data specifies the start of designating the members of the Cluster1.

[0266]

In the line 44 in FIG. 42, the feature data designates that the members of the Cluster1 are shots and whose names are "shot2, shot4, ...".

[0267]

Further in the line 45 in FIG. 42, the feature data specifies the end of designating the members of the Cluster1.

[0268]

Next in the line 46 in FIG. 42, the feature data designates the beginning of the Abstract descriptor to designate a representative part of the content of the Cluster1.

[0269]

In the line 47 in FIG. 42, the feature data specifies the Prototype descriptor. It is specified herein that the representative part of the content of the Cluster1 is "shot4".

[0270]

Further, in the lines 48 and 49 in FIG. 42, the feature data specifies the end of the Abstract descriptor and the end of the Cluster descriptor respectively.

[0271]

Next in the lines 50 and 51 in FIG. 42, the feature data specifies the Cluster

descriptors respectively, and declares that the names of the second and third Clusters are "Cluster2" and "Cluster3" respectively. Note that in such positions in the lines, there are specified many descriptors for describing the content. However, they are omitted herein.

[0272]

In the line 52 in FIG. 42, the feature data specifies the end of the Hierarchy descriptor.

[0273]

In the line 53 in FIG. 42, the feature data declares that the description of some AVHierarchys specifying the index of the scenes is to end.

[0274]

Further in the line 54 in FIG. 42, the feature data describes many descriptors specifying the content of the scene, but they will be omitted here.

[0275]

Next, in the line 55 in FIG. 42, the feature data declares that there are some AVGraph descriptors specifying the structure of the scene subsequently.

[0276]

In the line 56 in FIG. 42, the feature data specifies the beginning of the AVGraph descriptor to declare that the components of the AVGraph is SHOT_GROUP.

[0277]

In addition, in the lines 57 to 61 in FIG. 42, the feature data specifies the AVGraphEdge descriptor. In the line 57, it is specified that the AVGraph starts at Cluster1. This Cluster1 has been designated in the line 42.

[0278]

Further in the lines 58, 59 and 60 in FIG. 42, the feature data specifies that the AVGraph extends from the Cluster1 to Cluster2, from the Cluster3 to Cluster2 and from the Cluster2 to Cluster3, respectively.

[0279]

In the line 61 in FIG. 42, the feature data specifies that the AVGraph ends at Cluster3, and further specifies the end of the AVGraph descriptor in the line 62.

[0280]

In the line 63 in FIG. 42, the feature data declares that the description of some AVGraph descriptors specifying the scene structure is over.

[0281]

In the line 64 in FIG. 42, the feature data specifies many descriptors describing the content of the scene and then in the line 65, it further specifies the end of the descriptor indicating the scene description.

[0282]

As in the foregoing, the description scheme which describes the scene structure in a certain video data, can be described by means of sets of a plurality of descriptors, so that the feature data can be described.

[0283]

The transmitter 10 generates such feature data according to the video data by means of the feature generator 12, and transmits them along with the video data to the receiver 20. Thus, the receiver 20 having received the feature data together with the video data, can easily restore the video structure of the video data just by analyzing the video data based on the feature data and thus generate a storyboard for example by means of the retrieval engine 23.

[0284]

Therefore, the receiver 20 requires no highly advanced processing capability and large volume of computation resources for analysis of the video structure of video data. Thus the receiver 20 can be manufactured inexpensively and the memory used therein may be of a small capacity one.

[0285]

Without prior knowledge of the content structure of the video data, the user can easily and privately access to the video data and thus can carry out a browsing efficiently.

[0286]

As having been described in the foregoing, the data describing method according to the present invention, defined as the description scheme, is to realize an advanced method for retrieval of the content of video data for browsing. The feature data described using this data describing method can specify the video structure of

video data accurately, easily and efficiently. Also, since the descriptors are formed inheriting functions of descriptors from ones in the higher class, they can easily be extended to ones having higher functions to organize a highly extensible description scheme, and reduce capacity for the feature data. Thus, the memory used in the receiver 20 can be of a reduced capacity type.

[0287]

Note that the present invention is not limited to the embodiment having been described in the foregoing but each descriptor may not always be a one which inherits the functions from a description in the higher class, for example. In this case, however, it is necessary to newly prepare all descriptors in the higher class for one descriptor which inherits from the higher-class descriptors, for the feature data.

[0288]

Also, the present invention may of course adopt any other approach for processing to extract a video structure from video data than having been described in the foregoing.

[0289]

Thus, needless to say, the present invention may be modified appropriately without departing from the scope and spirit of the present invention.

[0290]

[Advantages of the Invention]

As having been described in detail above, a data describing method according

to the present invention, is one for describing feature data which specifies characteristics of contents of input data, wherein a descriptor as a unit element defining a describing method includes structure to have its attribute defined as a lower-level element, and may include a descriptor as an attribute.

[0291]

Therefore, a data describing method according to the present invention can describe the structured feature data using a set of descriptors which includes structure, and easily extract features of the content of input data based on the feature data.

[0292]

Also, a data processor according to the present invention is one for generating feature data which specifies characteristics of the content of input data, and includes means for generating feature data on the basis of a descriptor, that descriptor can be a unit element defining a describing method, includes structure to have its attribute defined as lower-level element and may include a descriptor as an attribute.

[0293]

Therefore, the data processor according to the present invention can generate structured feature data using a set of descriptors which includes structure, and represent the features of the content of input data accurately, easily and efficiently based on the feature data.

[0294]

Further the data processor according to the present invention is one for utilizing

feature data which specifies characteristics of the content of input data, and includes means for restoring a feature of the content of the input data by utilizing the feature data generated on the basis of a descriptor, that descriptor can be a unit element defining a describing method, includes structure to have its attribute defined as lower-level element and may include a descriptor as a attribute.

[0295]

Therefore, the data processor according to the present invention can restore the feature of the content of input data on the basis of the feature data, and extract the features of the content of input data without the necessity of any highly advanced processing capability and large volume of computation resources.

[Brief Description of the Drawings]

[FIG. 1]

An view explaining structure of the video data to which the present invention is applicable, also explaining modeled structure of video data.

[FIG. 2]

A block diagram explaining structure of the data transmitting-receiving system represented as an embodiment of the present invention.

[FIG. 3]

A table explaining a description format for a descriptor.

[FIG. 4]

A diagram explaining relation of inheritance between descriptors.

[FIG. 5]

A table explaining a description format for an AVelement descriptor.

[FIG. 6]

A table explaining a description format for an Avelement descriptor.

[FIG. 7]

A table explaining a description format for an Avabstract descriptor.

[FIG. 8]

A table explaining a description format for an Avprototype descriptor.

[FIG. 9]

A view explaining a prototype.

[FIG. 10]

A table explaining a description format for an Avhierarchy descriptor.

[FIG. 11]

A table explaining a description format for a Cluster descriptor.

[FIG. 12]

A view explaining a clustering.

[FIG. 13]

A view explaining a composite.

[FIG. 14]

A table explaining a description format for a Composite Segment descriptor.

[FIG. 15]

Views explaining a media structure graph.

[FIG. 16]

A table explaining a description format for an Avgraph descriptor.

[FIG. 17]

A table explaining a description format for an AVgraphEdge descriptor.

[FIG. 18]

A table explaining a description format for a Signature descriptor.

[FIG. 19]

A table explaining a description format for a Temporal Signature descriptor.

[FIG. 20]

A table explaining a description format for an activity-level-descriptor.

[FIG. 21]

A table explaining a description format for a Weight descriptor.

[FIG. 22]

A table explaining a description format for a WeightVector descriptor.

[FIG. 23]

A table explaining a description format for a Color descriptor.

[FIG. 24]

A table explaining a description format for a Color Space descriptor.

[FIG. 25]

A table explaining a description format for a concept-descriptor.

[FIG. 26]

A table explaining a description format for a Color Signature descriptor.

[FIG. 27]

A table explaining a description format for a Synopsis descriptor.

[FIG. 28]

A table explaining a description format for an audio frame descriptor.

[FIG. 29]

A table explaining a description format for a visual frame descriptor.

[FIG. 30]

A table explaining a description format for a Browsing Node descriptor.

[FIG. 31]

A table explaining a description format for a Shot descriptor.

[FIG. 32]

A table explaining a description format for a Scene descriptor.

[FIG. 33]

A table explaining a description format for a Topic descriptor.

[FIG. 34]

A table explaining a description format for a Program descriptor.

[FIG. 35]

A table explaining a description format for a Highlight descriptor.

[FIG. 36]

A flow chart explaining a series of processes effected in detecting and grouping a scene in the transmitter of the data transmitting-receiving system.

[FIG. 37]

A view explaining a dynamic feature sampling processing in the transmitter of the data transmitting-receiving system.

[FIG. 38]

A view explaining a dissimilarity threshold.

[FIG. 39]

A view explaining a temporal threshold.

[FIG. 40]

A flow chart explaining a series of processes effected in grouping segments in the transmitter of the data transmitting-receiving system.

[FIG. 41]

A view explaining an example of a description scheme.

[FIG. 42]

A view explaining an example of feature data.

[Explanation of Reference Symbols]

10 Transmitter

11 Memory part

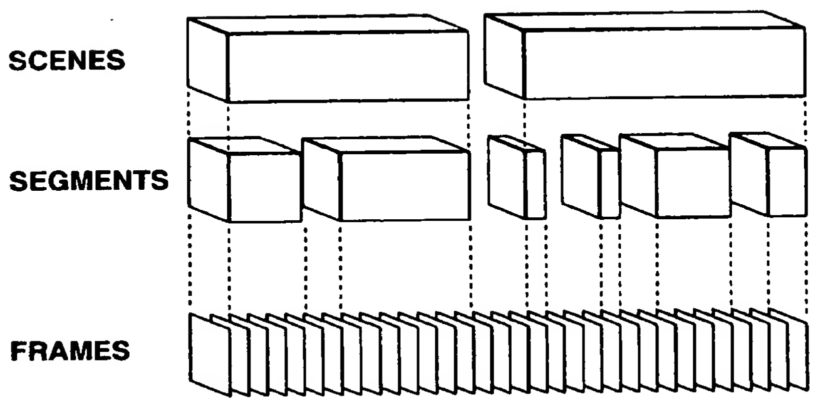
12 Feature data generator

20 Receiver

23 Retrieval engine

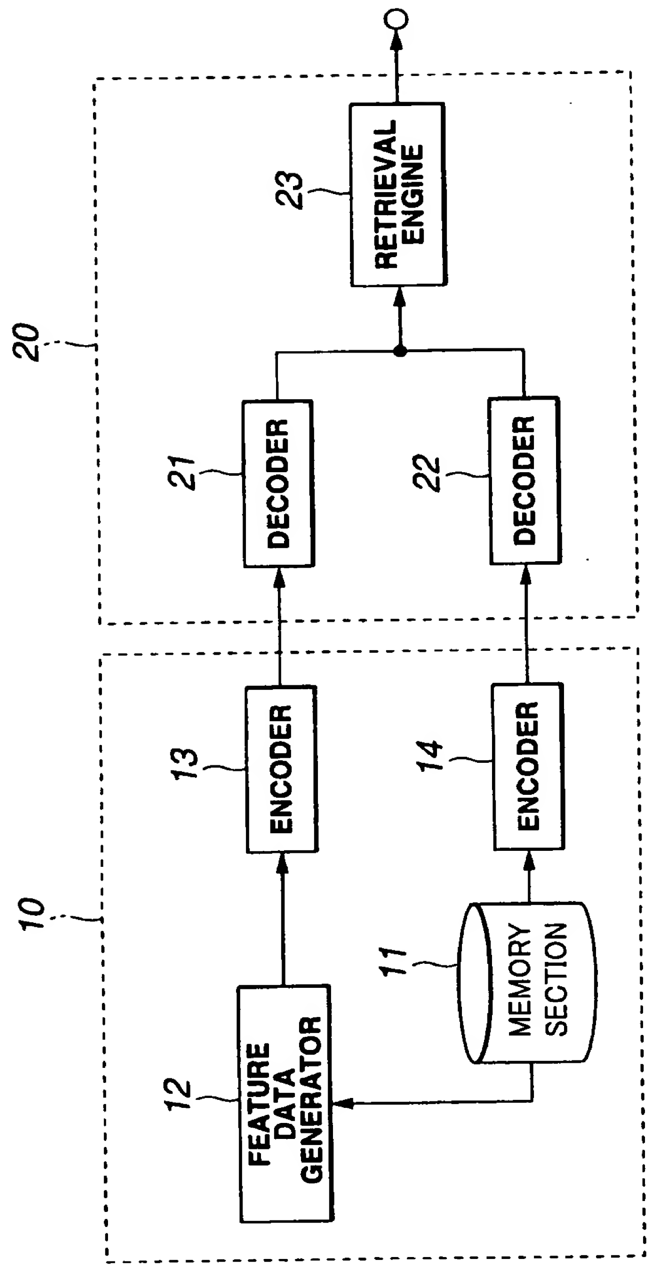
[DOCUMENT NAME] DRAWING

[FIG. 1]



VIDEO STRUCTURE HIERARCHY MODEL

[FIG. 2]



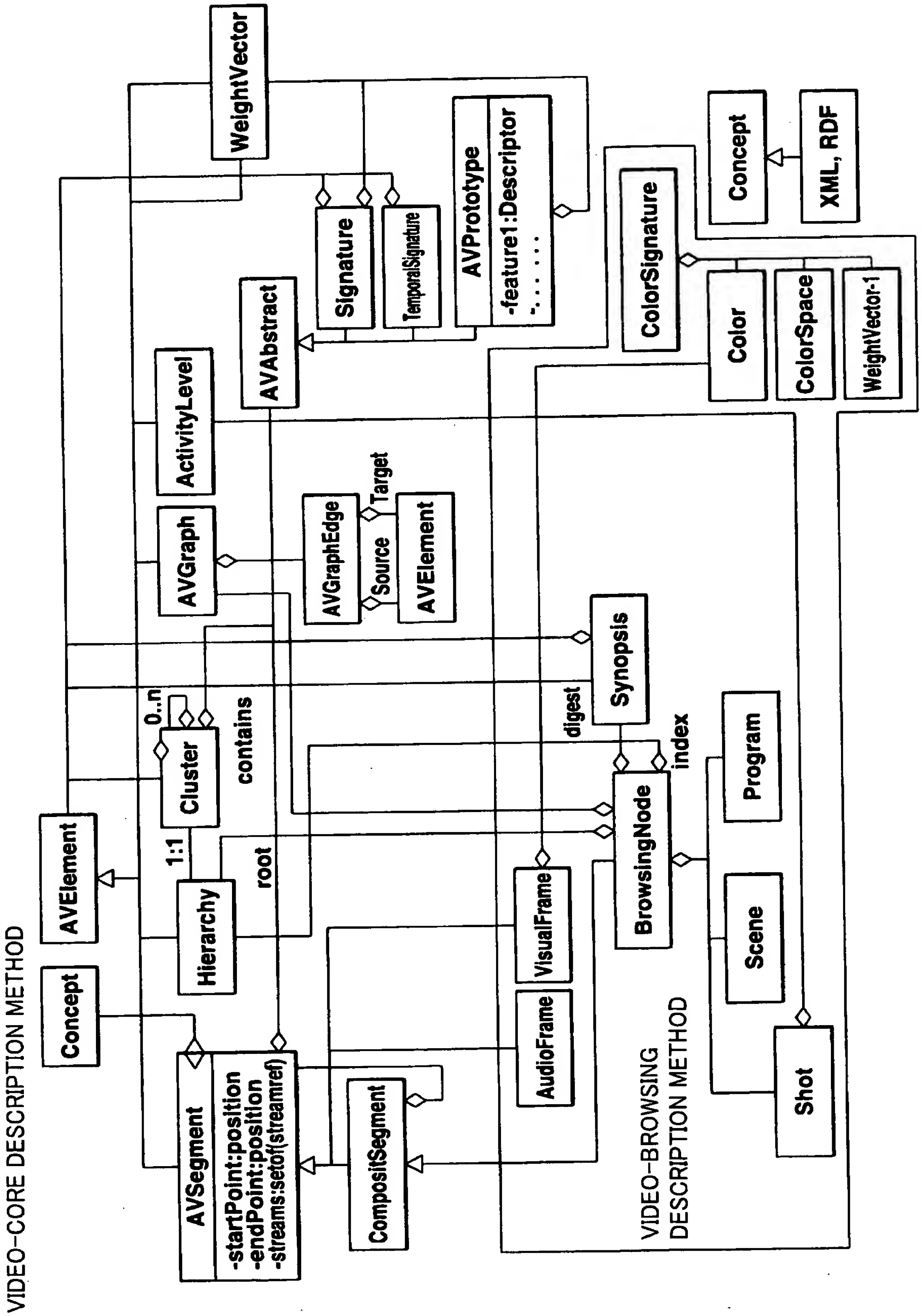
STRUCTURAL BLOCK DIAGRAM OF DATA TRANSMITTIMG-RECEIVING PROCESSING SYSTEM

[FIG. 3]

Descriptor		Superclass
Feature :	Type	Example Value
⋮	⋮	⋮

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 4]



VIEW EXPLAINING RELATION OF INHERITANCE BETWEEN DESCRIPTORS

[FIG. 5]

Descriptor : AVElement (AVE)		**Superclass : None**
Feature :	**Feature : Type**	**Example Value**

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 6]

Descriptor : AVSegment (AVS)		**Superclass : AVElement**
Feature :	**Feature Type**	**Example Value :**
StartTime	Timecode	
EndTime	Timecode	
Streams	setof(streamref)	
Abstract	setof(AVAbstract)	**Key frame**
Concept	setof(Concept)	**An RDF structure describing the content of this segment**

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 7]

Descriptor : AVAbstract (ABST)		Superclass : AVElement
Feature :	**Feature Type**	**Example Value :**

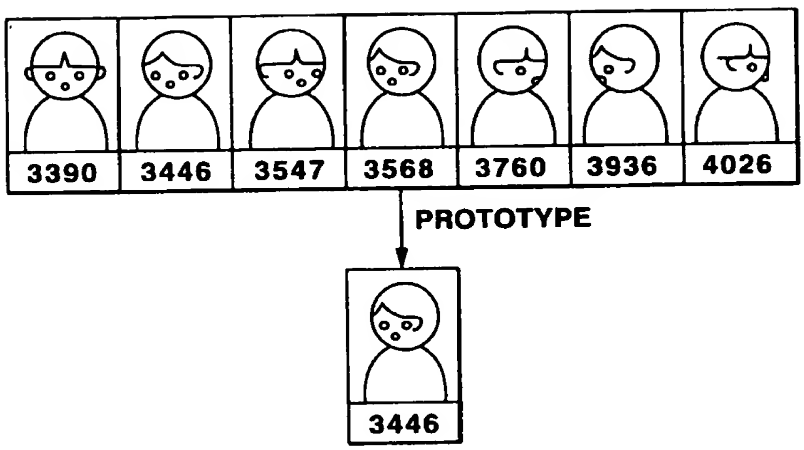
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 8]

Descriptor : AVPrototype (AVP)		Superclass : AVAbstract
Feature :	**Feature Type**	**Example Value :**
Kind	String	"Shot" "Scene" "VisualFrame"
Weights	WeightVector	
Feature1	Descriptor1	
Feature2	Descriptor2	
...	...	...
Featuren	Descriptorn	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 9]



VIEW EXPLAINING PROTOTYPE

[FIG. 10]

Descriptor : AV **Hierarchy** (HIER)		**Superclass :** **AVElement**
Feature :	**Feature Type**	**Example Value :**
Root	**Cluster**	
Kind	**String**	

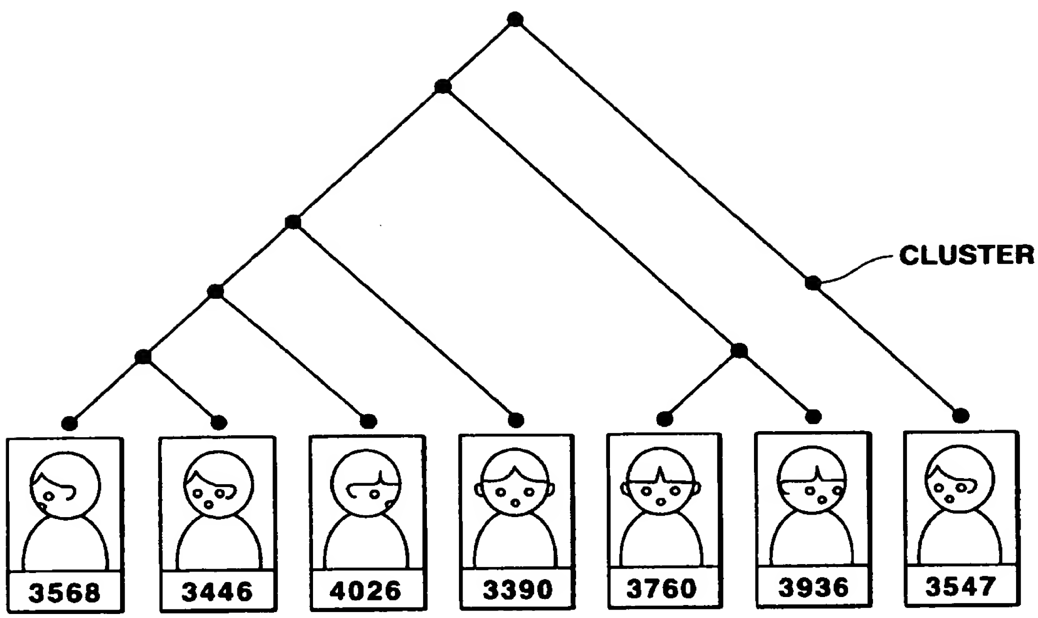
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 11]

Descriptor : **Cluster** (CN)		**Superclass :** **AVElement**
Feature :	**Feature Type**	**Example Value :**
Members	setof(**AVElement**)	
Abstract	**AVAbstract**	

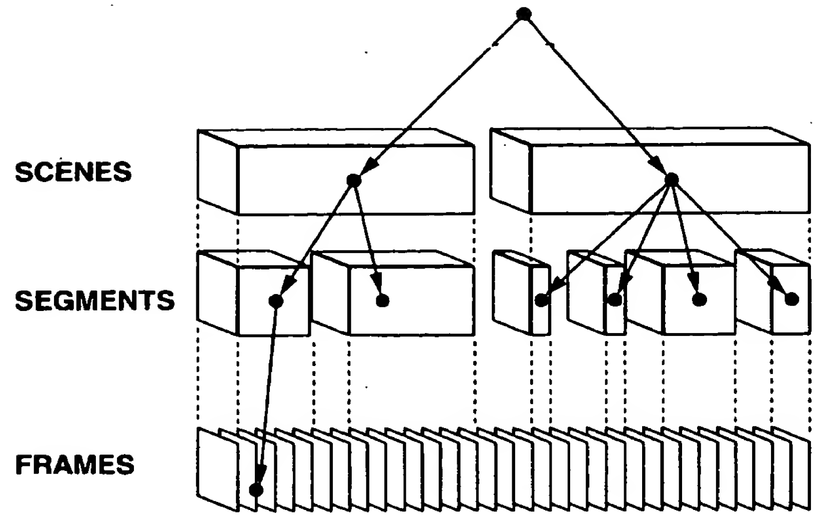
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 12]



VIEW EXPLAINING CLUSTERING

[FIG. 13]



VIEW EXPLAINING COMPOSITE

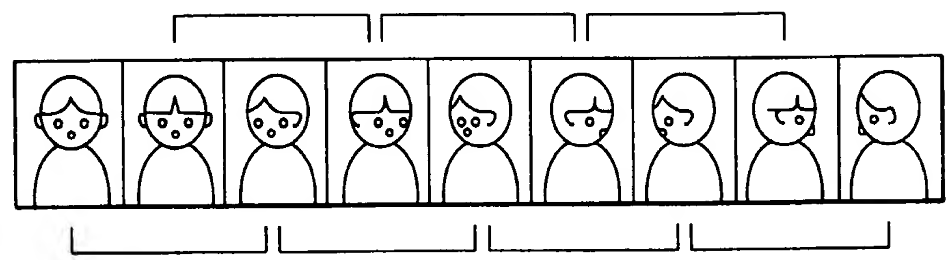
[FIG. 14]

Descriptor: CompositeSegment	Superclass: AVSegment	
Feature:	Feature Type	Example Value:
Components	setof (AVSegment)	

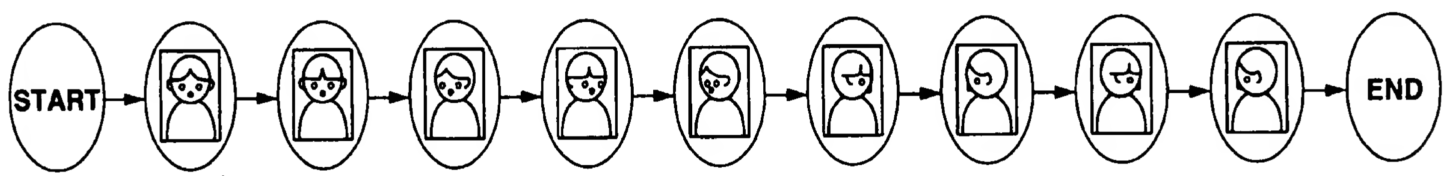
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 15]

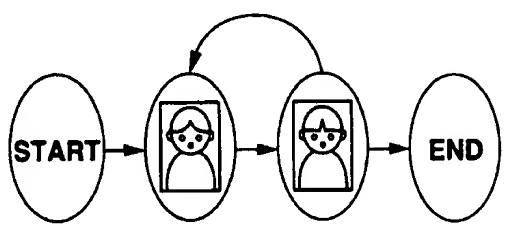
(A)



(B)



(C)



VIEW EXPLAINING MEDIA STRUCTURE GRAPH

[FIG. 16]

Descriptor : AVGraph (AVG)		Superclass : AVElement
Feature :	Feature Type	Example Value :
Edges	setof (AVGraphEdge)	
Kind	String	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 17]

Descriptor : AVGraphEdge (AVEDGE)		Superclass : None
Feature :	Feature Type	Example Value :
Source	AVElement	
Target	AVElement	
Label	String	"follows" "similarTo"

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 18]

Descriptor : Signature(SIG)		**Superclass :** **AVAbstract**
Feature :	**Feature Type**	**Example Value :**
Weights	**WeightVector** (n)	
Members	AVElement(n)	
Kind	**String**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 19]

Descriptor : **TemporalSignature** (TS)		**Superclass :** **AVAbstract**
Feature :	**Feature Type**	**Example Value :**
Time	**WeightVector**	
Members	Setof(AVElement)	
Kind	**String**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 20]

Descriptor : **ActivityLevel** (ACT)		Applies to AVElement
Feature :	**Feature Type**	**Example Value :**
ActivityLevel	**Real**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 21]

Descriptor : Weight(WGT)	Applies to AVElement
Feature :	**Feature Type**
Weighting	**Real**

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 22]

Descriptor: WeightVector(WV)		Applies to AVElement
Feature :	**Feature Type**	**Example Value :**
Weights	Weights(n)	
normalized	Boolean	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 23]

Descriptor : Color(COR)	Applies to: AVElement	Super type: None
Feature :	**Feature Type**	**Example Value :**
Color	Real(3)	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 24]

Descriptor : ColorSpace	Applies to: AVElement	Super type : None
Feature :	**Feature Type**	**Example Value :**
ColorSpace	**String**	**"HSV"**

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 25]

Descriptor : Concept (CPT)	Applies to AVElement	**Superclass : None**
Feature :	**Feature Type**	**Example Value :**
Concept	**RDFNode or XML or . . .**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 26]

Descriptor : ColorSignature(CHT)	Applies to: AVElement	**Super type : None**
Feature :	**Feature Type**	**Example Value :**
Colorspace	Colorspace	"RGB"
Colors	Color(n)	
Weights	WeightVector	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 27]

Descriptor : Synopsis		Superclass : AVElement
Feature : Concept (CPT)	Feature Type	Example Value :
Kind	String	SLIDE SHOW
Sequence	List of(AVElement)	
IsOrdered	Boolean	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 28]

Descriptor : AudioFrame(AF)	Superclass : AVSegment
Feature :	Feature Type

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 29]

Descriptor : VisualFrame (VF)		Superclass : AVSegment
Feature :	Feature Type	Example Value :
Color	ColorSignature	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 30]

Descriptor : BrowsingNode (BRO)		Superclass : CompositeSegment
Feature :	Feature Type	Example Value :
Index	setof (AVHierarchy)	A classification of the shots in a video
Synopsis	setof (AVSynopsis)	Sequence of key frames
Structure	AVGraph (optional)	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG. 31]

Descriptor : Shot (SHT)		Superclass : BrowsingNode
Feature :	Feature Type	Example Value :
Activity	ActivityLevel	"0.5"
Abstract	From AVSegment	
Synopsis	From BrowsingNode	
Index	From BrowsingNode	
Structure	From BrowsingNode	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG.32]

Descriptor : Scene (SCN)		Superclass : BrowsingNode
Feature :	**Feature Type**	**Example Value :**
Kind	**String**	**"DIALOG" "COMMERICAL" "ACTION" "ANCHOR"**
Abstract	**From AVSegment**	
Synopsis	**From BrowsingNode**	
Index	**From BrowsingNode**	
Structure	**From BrowsingNode**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG.33]

Descriptor : Topic(TPC)		Superclass : BrowsingNode
Feature :	**Feature Type**	**Example Value :**
Abstract	**From AVSegment**	
Synopsis	**From BrowsingNode**	
Index	**From BrowsingNode**	
Structure	**From BrowsingNode**	

DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG.34]

Descriptor : Program (PGM)		Super type : BrowsingNode
Feature :	**Feature Type**	**Example Value :**
Abstract	**From AVSegment**	Kind: TOPIC ABSTRACT SCENE ABSTRACT SHOT ABSTRACT
Synopsis	**From BrowsingNode**	
Index	**From BrowsingNode**	
Structure	**From BrowsingNode**	

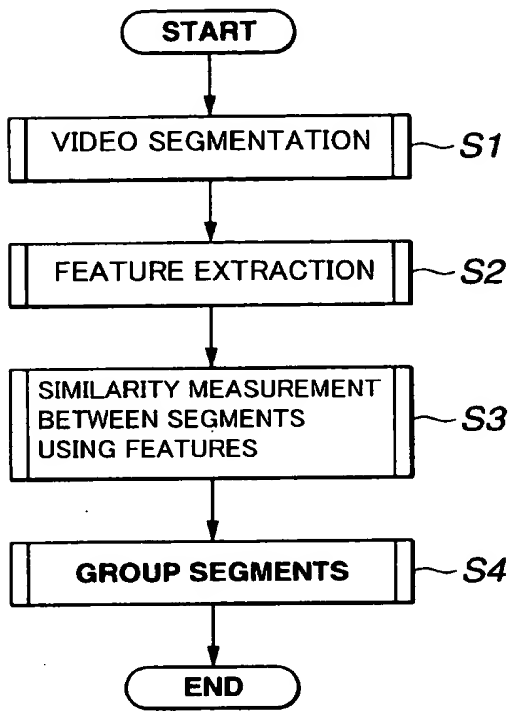
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG.35]

Descriptor: Highlight(HI)		Superclass: BrowsingNode
Feature :	**Feature Type**	**Example Value :**
Interest	Weight	

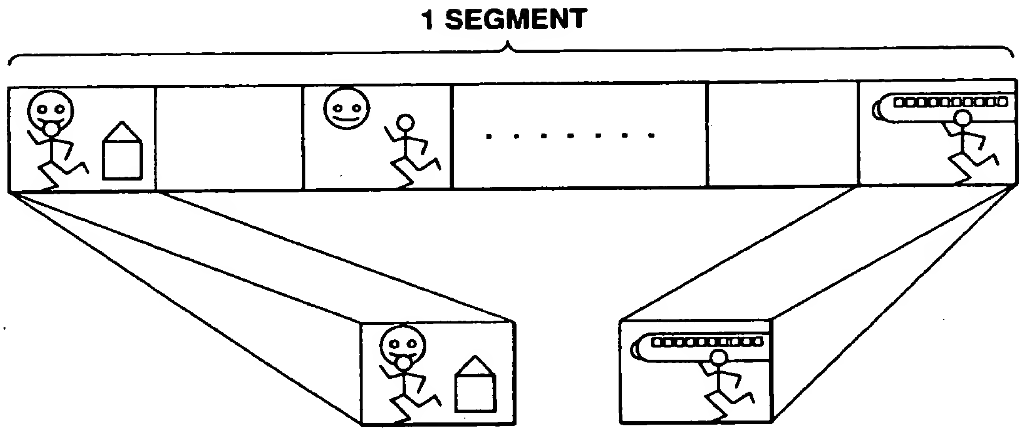
DESCRIPTION FORMAT FOR DESCRIPTOR

[FIG.36]



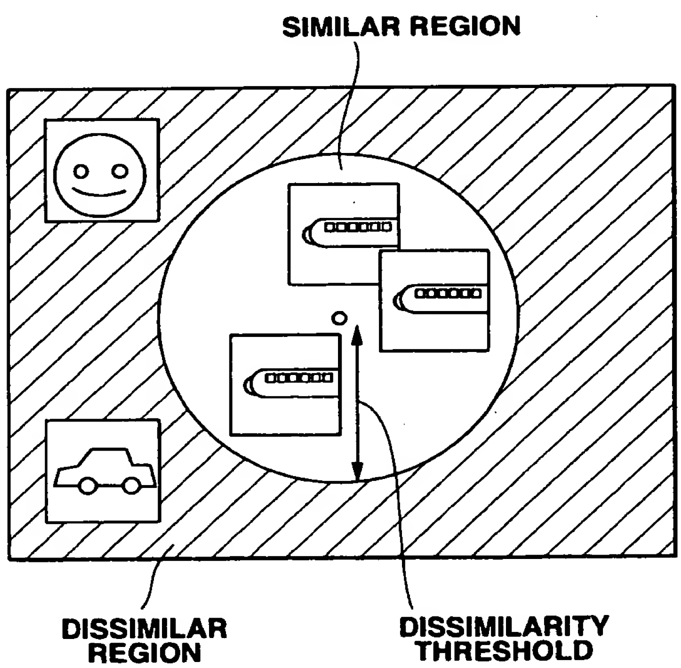
A SERIES OF PROCESSING IN TRANSMITTER

[FIG.37]



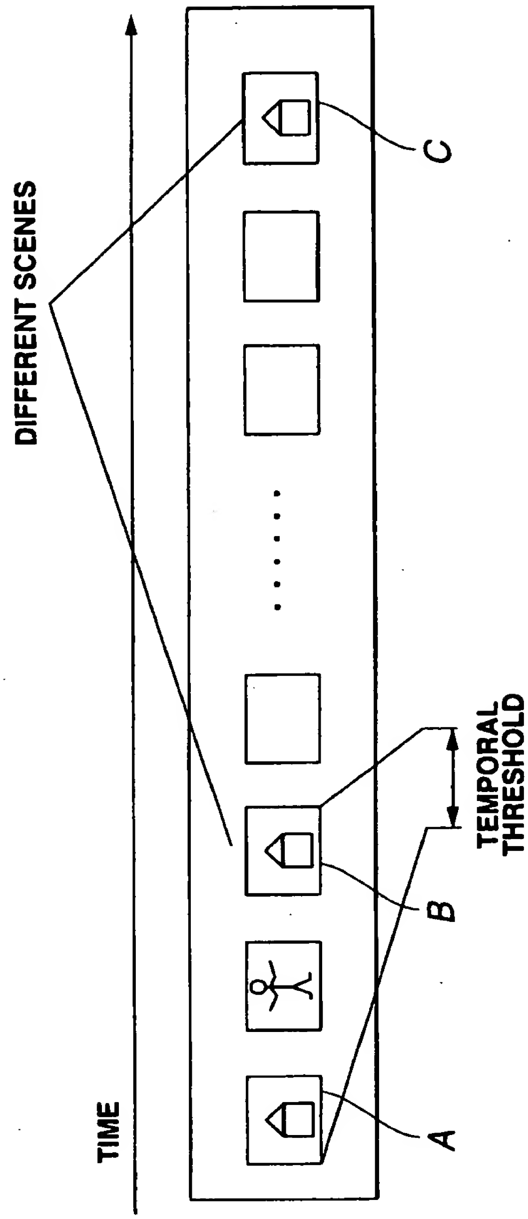
VIEW EXPLAINING DYNAMIC FEATURE SAMPLING METHOD

[FIG.38]



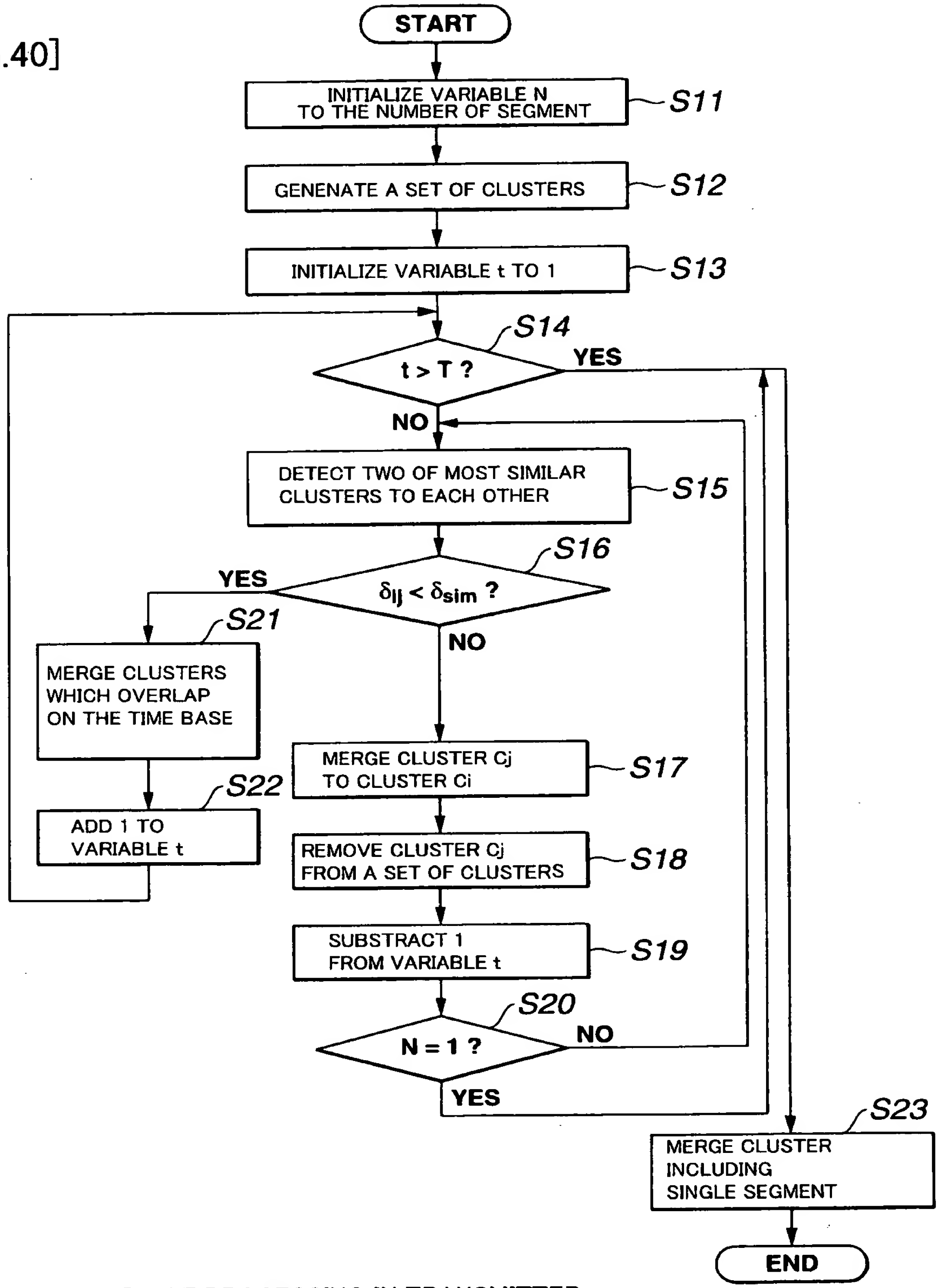
VIEW EXPLAINING DISSIMILARITY THRESHOLD

[FIG.39]



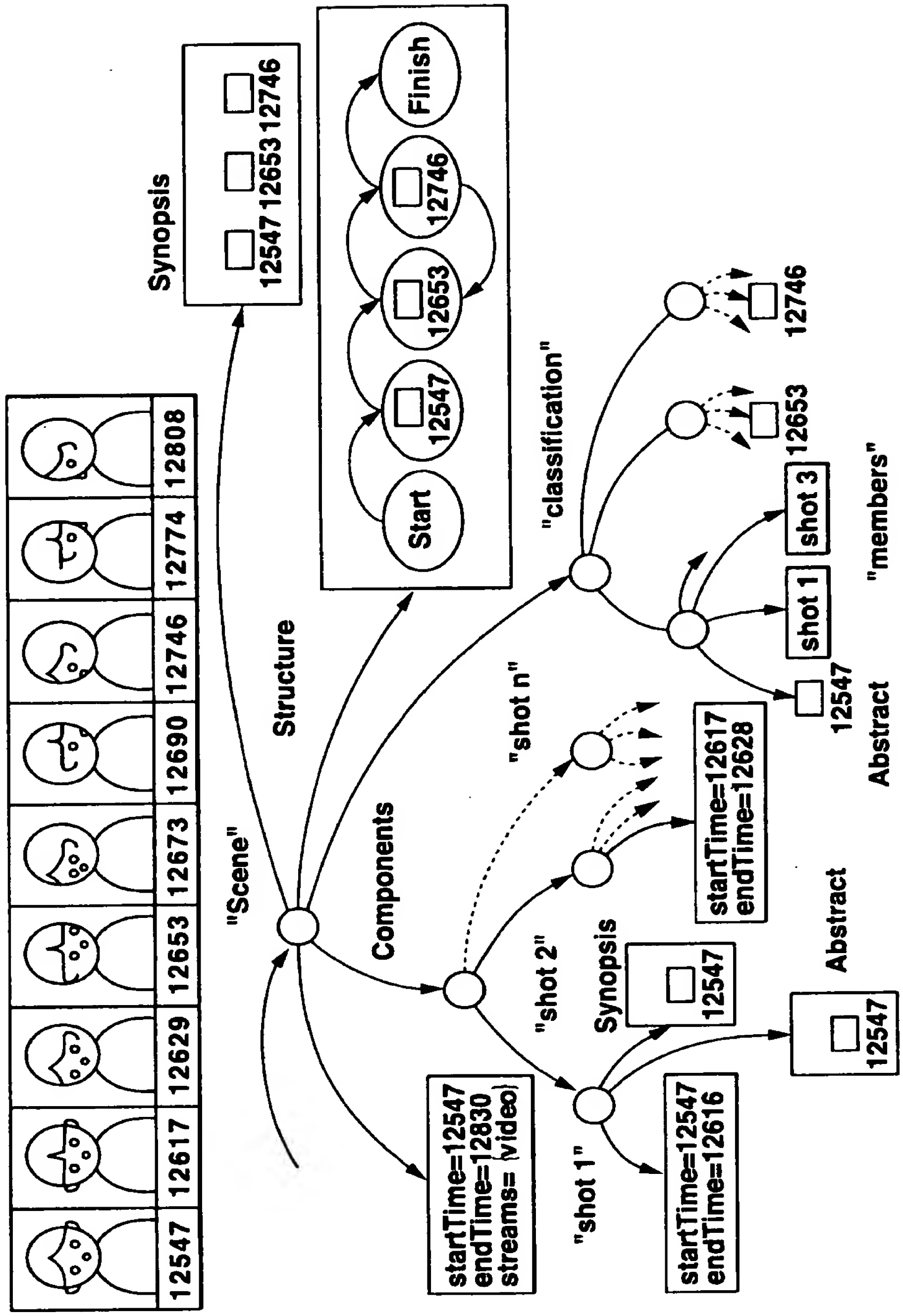
VIEW EXPLAINING TIME THRESHOLD

[FIG.40]



A SERIES OF PROCESSING IN TRANSMITTER

[FIG.41]



AN EXAMPLE OF DESCRIPTION SCHEME

[FIG.42]

```
<scene>
        <startPoint>12547</startPoint>                  ;AVSegment
        <endPoint>12830</endPoint>                      ;AVSegment
        <streams>{video}</streams>                      ;AVSegment
        <synopsis kind=KEY_FRAME>                       ;BrowsingNode
                <sequence>
                        < KEY_FRAME refid= KEY_FRAME 1>
                        < KEY_FRAME refid= KEY_FRAME 2>
                        < KEY_FRAME refid= KEY_FRAME 3>
                </sequence>
        </synopsis>
        <abstract>                                      ;AVSegment
                <signature kind=SHOT_SIGNATURE>
                        <members>
                                <shot refid=shot1> <shot refid=shot2>
                                <shot refid=shot3>
                        </members>
                        <weights>0.1 0.45 0.45</weights>
                </signature>
        </abstract>
        <components>                            ;CompositeSegment
                <shot id=shot1                                  ;Shot
                        <startPoint>12547</startPoint>          ;AVSegment
                        <endPoint>12616</endPoint>                      ;AVSegment
                        <abstract>                                      ;AVSegment
                                <signature kind=FRAME_SIGNATURE>
                                        <member><frame refid=12590></members>
                                        <weights>1.0</weights>
                                </signature>
                        </abstract>
                </shot>
                <shot id=shot2>
                        <startPoint>12617</startPoint>
                        <endPoint>12628</endPoint>
                        ...
                </shot>
                ...
                <shot id=shotn>...</shot>
        </components>
        <index>                                                 ;BrowsingNode
                <hierarchy kind=SHOT_INDEX>                     ;Hierarchy
                        <cluster id=cluster1>
                                <members>
                                <shot refid=shot2> <shot refid=shot4>...
                                </members>                      ;Cluster
                                </abstract>
                                        <prototype>shot 4</prototype>
                                </abstract>
                        </cluster>
                        <cluster id=cluster2>...</cluster>
                        <cluster id=cluster3>...</cluster>
                </hierarchy>
        </index>
        ...
        <structure>                                             ;BrowsingNode
                <avgraph kind=SHOT_GROUP>                       ;AVGraph
                        <avgraphedge source=start target=cluster1>
                        <avgraphedge source=cluster1 target=cluster2>
                        <avgraphedge source=cluster3 target=cluster2>
                        <avgraphedge source=cluster2 target=cluster3>
                        <avgraphedge source=cluster3 target=finish>
        </avgraph>
        <structure>
        ...
</scene>
```

AN EXAMPLE OF FEATURE DATA

[Document] Abstract

[Abstract]

[Object]

To realize a higher advanced method for retrieving contents of various video data, thereby to and browse.

[Means for Achieving the Object]

A descriptor becoming a unit element defining a describing method for describing feature data which specifies the feature of contents of various video data, includes structure to have its attribute defined as lower-level elements, and is formed by a description format which is able to include other descriptors as attributes. Namely, the description format for a descriptor comprises Feature specifying a descriptor of the lower-level element or a type of numerical value, Feature Type specifying a descriptor of the lower-level element or a type of numerical value, and Example Value specifying a descriptor of the lower-level element or an example of numerical value.

[Elected Figure] FIG. 3

Information of Record for Applicant

Identification Number:	[000002185]
1. Date of Change:	August 30, 1990
[Reason of Change]	Registration
[Address]	7-35, Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan
[Name]	Sony Corporation